Cryptanalyse physique de circuits cryptographiques à l'aide de sources LASER

Cyril Roscian

Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

NNT : 2013 EMSE 0708

Thèse

présentée par

Cyril ROSCIAN

Pour obtenir le grade de
Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : **Microélectronique**

Cryptanalyse physique de circuits cryptographiques à l'aide de sources LASER

Soutenue à Gardanne le 8 Octobre 2013

Président du jury
M. Régis LEVEUGLE ..... Laboratoire TIMA, Grenoble

Examinateurs
M. Hervé CHABANNE ..... Morpho, Paris

Rapporteurs
M. Bruno ROUZEYRE ..... LIRMM, Montpellier
M. Sylvain GUILLEY ..... TELECOM-ParisTech, Paris

Directeur de thèse
Mme. Assia TRIA ..... CEA-TECH/CMP, Gardanne

Encadrant
M. Jean-Max DUTERTRE ..... ENSM-SE/CMP, Gardanne

Invité
M. Christophe GIRAUD ..... Oberthur Technologies, Bordeaux

Spécialités doctorales :	Responsables :
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche
MECANIQUE ET INGENIERIE	S. Drapier, professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche
MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
INFORMATIQUE	O. Boissier, Professeur
IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
GENIE INDUSTRIEL	A. Dolgui, Professeur
MICROELECTRONIQUE	

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BENABEN	Patrick	PR1	Sciences et génie des matériaux	CMP
BERNACHE-ASSOLLANT	Didier	PR0	Génie des Procédés	CIS
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR1	Informatique	FAYOL
BORBELY	Andras	MR(DR2)	Sciences et génie de l'environnement	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BURLAT	Patrick	PR2	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DARRIEULAT	Michel	IGM	Sciences et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR1	Sciences et génie des matériaux	SMS
DESRAYAUD	Christophe	PR2	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FEILLET	Dominique	PR2	Génie Industriel	CMP
FOREST	Bernard	PR1	Sciences et génie des matériaux	CIS
FORMISYN	Pascal	PR0	Sciences et génie de l'environnement	DIR
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GIRARDOT	Jean-jacques	MR(DR2)	Informatique	FAYOL
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HAN	Woo-Suck	CR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
INAL	Karim	PR2	Microélectronique	CMP
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Franck	DR	Sciences et génie des matériaux	SMS
PERIER-CAMBY	Laurent	PR2	Génie des Procédés	DFG
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	CR	Génie des Procédés	CIS
ROUSTANT	Olivier	MA(MDC)		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
SZAFNICKI	Konrad	MR(DR2)	Sciences et génie de l'environnement	CMP
TRIA	Assia		Microélectronique	CMP
VALDIVIESO	François	MA(MDC)	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	MR(DR2)	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Informatique	CIS

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	MCF	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	MCF	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 07/04/2013

PR 0	Professeur classe exceptionnelle
PR 1	Professeur 1ère classe
PR 2	Professeur 2ème classe
PU	Professeur des Universités
MA (MDC)	Maître assistant
DR	Directeur de recherche

Ing.	Ingénieur
MCF	Maître de conférences
MR (DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
IGM	Ingénieur général des mines

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
FAYOL	Institut Henri Fayol
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

À mon parrain

Table des matières

Table des figures

Liste des tableaux

Introduction

La société actuelle est résolument tournée vers l'échange d'informations. Le développement accru des moyens de communication (smartphones, box internet, etc.) mais aussi des moyens de paiement électronique (carte de paiement, PayTV, etc.) a posé de nouvelles problématiques aux fabricants de circuits intégrés pour assurer une transmission sécurisée des informations ou sécuriser les moyens de paiements. La cryptographie s'est alors imposée comme un outil répondant aux exigences posées par ces nouveaux moyens de communication ou de paiement.

Le dispositif le plus répandu utilisant des algorithmes cryptographiques est la carte à puce. Celui-ci, utilisé aussi bien dans les télé-communications avec les cartes SIM que dans le milieu bancaire avec les cartes de paiement électronique, embarque le plus souvent un circuit intégré ayant une partie dédiée aux calculs cryptographiques. Ces circuits vont donc manipuler des données sensibles et doivent être protégés. Selon le principe de Kerckhoffs, la sécurité de données chiffrées doit dépendre uniquement de la clef de chiffrement utilisée, et non de l'algorithme ou des paramètres de celui-ci. C'est donc cette clef qu'il faut protéger d'un attaquant potentiel.

Malgré la robustesse mathématique des différents algorithmes de chiffrement utilisés, les circuits intégrés sont cependant vulnérables aux attaques matérielles. Une partie de celles-ci vise à perturber le fonctionnement normal du circuit intégré lors d'un calcul

cryptographique. La perturbation du circuit provoque une erreur sur le résultat du chiffrement. Ces résultats obtenus sont ensuite analysés afin de retrouver la clef sécrète. On parlera alors d'injection de fautes provoquant une erreur de calcul. Ces attaques reposent dans la majorité des cas sur la capacité de l'attaquant à injecter des fautes à des instants précis du calcul cryptographique. L'efficacité de ces attaques peut dépendre aussi du modèle de fautes injectées. Il est donc important de maîtriser le modèle de fautes.

Plusieurs moyens de perturbation existent, nécessitant parfois une préparation préalable du circuit visant à ôter une partie du boîtier laissant un accès direct au circuit intégré. L'utilisation d'un tir laser est un des moyens les plus efficaces d'injection permettant d'avoir une précision spatiale du tir sur le circuit intégré mais aussi une précision temporelle pour synchroniser le tir laser avec l'exécution du calcul cryptographique. Le laser est souvent considéré comme un moyen d'injection permettant d'obtenir les fautes nécessaires aux différentes attaques. Cependant, les modèles de fautes induits par ce moyen d'injection n'ont pas été clairement caractérisés de même que l'exploitation éventuel de ces modèles de fautes.

Cette thèse a donc pour objectif d'étudier les modèles de fautes possibles lors d'injection de fautes par tir laser, d'exploiter ces modèles de fautes pour évaluer la menace liée à ces attaques en fautes mais aussi de tester la robustesse des circuits cryptographiques embarquant des protections contre les attaques en fautes.

Le premier chapitre de cette thèse présentera les différents mécanismes permettant l'injection de fautes par tir laser mais aussi l'influence des paramètres du laser sur l'injection de fautes. Puis, le second chapitre portera sur l'étude des modèles de fautes possibles lors d'injection de fautes sur une cellule SRAM mais aussi sur la mémoire RAM d'un micro-contrôleur. Les différents résultats expérimentaux seront accompagnés de simulations électriques d'injection de fautes par laser pour mieux comprendre les modèles de fautes observés. Enfin le dernier chapitre abordera les modèles de fautes induits par tir laser sur un circuit intégré implémentant l'algorithme de chiffrement symétrique AES. À l'aide des données collectées ainsi que des informations sur les modèles de fautes possibles, un schéma d'attaque en fautes sera amélioré, tirant parti des modèles de fautes observés. La dernière partie de ce chapitre évaluera la robustesse de ce circuit intégré

embarquant des dispositifs de protection contre l'injection de fautes, à l'aide des modèles de fautes observés.

Chapitre

1 La cryptographie et l'injection de fautes par laser

Préambule

Le but de ce chapitre est de rappeler succinctement les bases de la cryptographie ainsi que les différents types d'attaques matérielles possibles. Il sera ensuite présenté un état de l'art des mécanismes d'injections de fautes par laser sur circuit CMOS *mais aussi de l'influence des différents paramètres d'un tir laser sur l'injection de fautes.*

Contents

1.1 La cryptographie

1.1.1 Introduction

La cryptographie est une discipline permettant de transmettre des informations de manière confidentielle. Pour rendre les informations à transmettre confidentielles, celles-ci subissent une transformation appelée chiffrement, rendant le texte clair chiffré incompréhensible pour tout le monde. Depuis l'antiquité, ces méthodes de chiffrement ont été utilisées principalement pour la transmission d'informations militaires. Jusqu'au $XIX^{ème}$ siècle, ces méthodes de chiffrement étaient basées sur deux grands principes : la **permutation**, permettant d'avoir une bonne diffusion de l'information du texte clair vers le texte chiffré, et la **substitution**, permettant d'avoir une bonne confusion (complexité de la relation entre le message chiffré et le texte clair). Pour ces méthodes de chiffrement, le secret, permettant de chiffrer et déchiffrer les données, est constitué de l'algorithme de chiffrement lui-même. Une des méthodes de chiffrement les plus connues et se basant sur ces principes est le chiffrement de *César* [24], consistant à effectuer un décalage de 3 lettres pour chaque lettre de l'alphabet. La notion de clef de chiffrement fut introduite au $XVI^{ème}$ avec le chiffre de *Vigenère* [24]. Pour cette algorithme, le secret repose sur la clef de chiffrement et peut être vue comme un mot de passe.

À la fin du $XIX^{ème}$, Auguste Kerckhoffs énonça les règles de ce qui allait devenir la cryptographie moderne [25]. Ces règles sont regroupées en six principes fondamentaux :

- Le système doit être matériellement, sinon mathématiquement, indéchiffrable.
- Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi.
- La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.
- Il faut qu'il soit applicable à la correspondance télégraphique.
- Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.
- Il est nécessaire que le système soit d'un usage facile et ne demande pas la connaissance d'une longue série de règles à observer.

De ces six principes, on retiendra surtout le deuxième énonçant le fait d'avoir un

algorithme public connu de tous, le secret résidant seulement dans la clef de chiffrement. De nos jours, la cryptographie peut être divisée en deux grandes familles, le chiffrement asymétrique (à clef publique) et le chiffrement symétrique (à clef secrète).

1.1.2 Chiffrement asymétrique

Le principe du chiffrement asymétrique fut présenté pour la première fois en 1976 par Diffie et Hellman [12]. Ce type de chiffrement consiste à utiliser des clefs différentes pour le chiffrement et le déchiffrement. La clef de chiffrement est publique et peut donc être distribuée sans risque de révéler le secret. En revanche le déchiffrement ne peut se faire qu'avec une clef secrète appelée clef privée. Ce type d'algorithme de chiffrement est principalement utilisé pour la distribution de clefs secrètes pour effectuer le chiffrement de données avec un algorithme de chiffrement symétrique.

Un des algorithmes de chiffrement à clef publique le plus connu et facile à mettre en œuvre est le RSA [48] du nom de ses concepteurs (Ron Rivest, Adi Shamir et Leonard Adleman). En 1985, l'utilisation de courbes elliptiques fut introduite par Victor Miller [38] et Neal Koblitz [27] pour réaliser du chiffrement à clef publique permettant l'utilisation de clefs plus courtes.

1.1.3 Chiffrement symétrique

Le chiffrement à clef secrète utilise la même clef pour le chiffrement et le déchiffrement des données. La clef de chiffrement doit donc être connue seulement des personnes autorisées à chiffrer ou déchiffrer les données confidentielles. On peut séparer les algorithmes de chiffrement symétriques en deux catégories : les chiffrements par flot et les chiffrements par bloc.

Chiffrement par flot

Le chiffrement par flot effectue un chiffrement bit à bit des données à chiffrer après une phase d'initialisation. Une suite de bits de chiffrement est fournie à partir d'un générateur de flux prenant en compte la clef secrète et un état interne déterminé pendant

la phase d'initialisation. A5/1, E0, RC4 font partie des algorithmes de chiffrement par flot les plus connus.

Chiffrement par bloc

Le chiffrement à clef secrète par bloc consiste dans un premier temps à diviser les données à chiffrer en blocs de n bits (généralement 32, 64, 128 ou 512 bits). Par la suite, chaque bloc est chiffré par l'algorithme à clef secrète. On obtient alors à la fin des blocs chiffrés de n bits constituant les données chiffrées. Les algorithmes de chiffrement par blocs les plus utilisés ont été standardisés par le NIST [40] (National Institute of Standard and Technology).

L'algorithme DES

L'algorithme DES a été standardisé en 1976 par le NIST [41]. Cet algorithme est basé sur le schéma de Feistel [34] et fonctionne avec des blocs de données de 64 bits et une taille de clef de 56 bits. Après une opération de permutation, les données subissent une transformation, répétée durant 16 itérations (*rondes*). Une permutation inverse est effectuée après la fin des 16 *rondes*. Durant chaque ronde, les données sont divisées en deux blocs de 32 bits puis échangés l'un avec l'autre selon un schéma de Feistel. En 1999, un algorithme utilisant plusieurs fois l'algorithme DES a été standardisé [41] et appelé 3DES. Le but de cet algorithme était d'avoir un algorithme de chiffrement symétrique plus robuste que le DES. Cet algorithme utilise 3 clefs de chiffrements différentes (K_1, K_2 et K_3) de 56 bits chacune et une taille de données de 64 bits. Un premier chiffrement DES des données est effectué avec K_1 puis les données chiffrées sont déchiffrées avec K_2 et enfin elles subissent un dernier chiffrement avec K_3.

L'algorithme AES

Les puissances de calculs des ordinateurs et la taille réduite de la clef de chiffrement utilisée par l'algorithme DES, font que celui-ci n'est plus sûr. Le NIST a donc lancée un concours en 1997 pour définir un nouveau standard de chiffrement symétrique. En 2001, l'algorithme *Rijndael* fut désigné vainqueur et renommé AES (Advanced Encryption Standard) [42].

L'AES est basé sur les réseaux de substitution-permutation [63]. La taille des données manipulées est de 128 bits avec une taille de clef de 128, 192 ou 256 bits. Cet algorithme est composé de quatre transformations élémentaires (SUBBYTES, SHIFTROWS, MIXCOLUMNS et ADDROUNDKEY) répétées 10, 12 ou 14 fois selon la taille de clef utilisée ; chaque répétition est appelée *"Ronde"*. S'y ajoute une ronde initiale comprenant seulement la transformation ADDROUNDKEY. La ronde finale n'effectue pas la transformation MIXCOLUMNS. Une description plus détaillée de cette algorithme est donnée à la partie 3.2.

1.1.4 Cryptanalyse physique

Les systèmes cryptographiques utilisés sont implémentés en technologie CMOS. Bien que les algorithmes en eux-mêmes soient sûrs mathématiquement, le fait qu'ils soient implémentés physiquement à l'aide de circuits intégrés crée des grandeurs observables ou perturbables permettant de réaliser des attaques sur ces circuits afin de retrouver la clef secrète.

Il existe plusieurs façons d'attaquer un circuit pour retrouver la clef secrète. On peut classer ces attaques en trois catégories : les attaques non invasives, invasives et semi-invasives. Chaque catégorie peut alors être aussi décomposée en deux sous-catégories : attaques réalisées de façon passives ou actives. Les attaques non-invasives concernent les attaques ne nécessitant aucune préparation du circuit ou une quelconque altération de celui-ci. Principalement ce type d'attaque regroupe les attaques par observation (*Side Channels Analysis*) de la consommation d'énergie [28], de l'émanation électromagnétique [17] ou du temps d'exécution [29], considérées comme passives. Les attaques utilisant des pulses (glitchs) électromagnétiques [11], de tension [6] [61] ou d'horloge [2] sont aussi des attaques non invasives mais cette fois actives puisqu'elles modifient le comportement du circuit.

Les attaques invasives regroupent les attaques nécessitant la modification du circuit [30] (active) ou l'utilisation d'appareil de probing [20] [16] (passives) pour avoir directement accès aux données du circuit.

La dernière catégorie d'attaques rassemble les attaques nécessitant de retirer une partie du boîtier entourant le circuit sans modifier celui-ci comme l'injection de fautes

laser [62] (active) ou l'observation de point chaud et d'émission de photons [59] (passives).

Le tableau 1.1 résume les différents types d'attaques.

TABLE 1.1: Classification des attaques matérielles.

	Passives	Actives
Non Invasives	Consommation d'énergie [28] Temps d'exécution [29] Emanations electromagnétiques [17]	Glitch de tension [6] [61] Glitch d'horloge [2] Glitch electromagnétique [11]
Invasives	Probing [20] [16]	Modification du circuit [30]
Semi-Invasives	Emission de photons [59] Observation de point chaud	Injection de fautes laser [62] [1]

Parmi les moyens d'injection de fautes visant à perturber le comportement du circuit intégré, le tir laser se démarque par sa faculté à pouvoir obtenir une très bonne résolution spatiale sur le circuit cryptographique attaqué. En effet, la taille de spot d'un tir laser étant modifiable, lors d'un tir, seulement une partie réduite du circuit intégré peut être touchée. Cela permet d'obtenir des fautes localisées, contrairement aux glitchs ayant un effet global sur le circuit, qui ne permettent pas d'avoir un contrôle spatial de l'injection de la faute. Cette caractéristique fait du laser un moyen d'injection de fautes très efficace.

Néanmoins, il est important de connaître les effets d'un tir laser sur les circuits CMOS ainsi que les mécanismes d'injection d'une faute pour mieux analyser les fautes injectées et définir au mieux les paramètres du tir laser. Les deux parties suivantes s'attachent donc à décrire les mécanismes d'injection de fautes par tir laser ainsi que les effets des différents paramètres d'un tir laser sur l'injection de fautes.

1.2 Mécanisme d'injection de fautes par laser

1.2.1 Terminologie

Pour une meilleure compréhension de la suite du document, on définit ici plusieurs termes nécessitant des définitions claires et précises.

Transitoire de courant

On parlera de transitoire de courant lorsque lors d'un tir laser, un photo-courant est créé grâce à l'effet photo-électrique.

Transitoire de tension (ou SET pour Single Event Transient)

On parlera de transitoire de tension (ou SET) lorsque le niveau logique de la sortie d'un élément combinatoire change à la suite d'un tir laser. Ce changement est temporaire, lorsque le tir laser est fini, le niveau logique retrouve son état initial. Ce changement temporaire du niveau logique peut se propager au travers des différents éléments logiques en aval de l'élément combinatoire où le SET est créé.

Un transitoire de tension est provoqué par un transitoire de courant. En revanche un transitoire de courant n'entraîne pas systématiquement un transitoire de tension.

Événement unique (ou SEU pour Single Event Upset)

On désignera par le terme événement unique (ou SEU), un changement de l'état logique mémorisé par un élément mémoire (DFF, SRAM, ect.).

Effet d'événement singulier (ou SEE pour Single Event Effect)

L'effet d'événement singulier (ou SEE) regroupe les effets possibles d'un tir laser, et plus généralement, les effets de particules radiatives sur les circuits intégrés. On retrouve donc sous le terme SEE , les événements de types transitoire de courant, SET ou SEU.

Faute ou erreur

On parlera d'une erreur lorsque le résultat d'un chiffrement est différent de celui attendu, indifféremment de la cause de cette erreur. Dans notre cas, une faute est le résultat d'un tir laser provoquant un SEU ou la création d'un SET provoquant la mémorisation par un registre en aval d'un niveau logique erroné. On obtient alors une erreur sur le résultat du chiffrement. Le terme erreur est beaucoup plus général que le terme de faute, celui-ci étant restreint à une modification du comportement du circuit.

1.2.2 Effet Photo-électrique

Un tir laser permet de créer, grâce à l'effet photo-électrique, des paires électrons-trous dans un composant CMOS. Ces paires électron-trou seront générées par un tir laser à condition que l'énergie des photons soit supérieure à l'énergie de la bande interdite du silicium [19]. Cette condition impose alors une longueur d'onde maximale de 1,10 μm. En effet, pour qu'un électron puisse être arraché à la bande de valence vers la bande

de conduction, l'énergie transmise par les photons doit être supérieure à l'énergie de bande interdite (E_{bg}), pour le silicium on a $E_{bg} = 1,12\ eV$. La relation entre l'énergie des photons et la longueur d'onde donne la condition exprimée à l'équation 1.2.

$$E_{ph} > E_{bg} \tag{1.1}$$

$$E_{ph} = \frac{hc}{\lambda}, \tag{1.2}$$

où h est la constante de Planck, c la vitesse de la lumière (m/s) et λ la longueur d'onde du laser. Des équations 1.1 et 1.2, on en déduit alors la valeur maximale pour la longueur d'onde, donnée à l'équation 1.3 :

$$\lambda < \frac{hc}{E_{bg}} \tag{1.3}$$

En temps normal, ces paires électron-trou, générées par l'effet photo-électrique se recombinent sans perturber notablement le fonctionnement du circuit, mais sous certaines conditions, des événements singuliers (SEE) peuvent apparaître.

1.2.3 Génération d'un photo-courant

La génération d'un photo-courant, ou transitoire de courant, peut se manifester lorsque le tir laser traverse une jonction PN polarisée en inverse. En effet, au lieu de se recombiner de façon naturelle sans perturber le circuit, les paires électron-trou sont balayées dans des sens opposés sous l'effet du champ électrique de la zone de charge espace (ZCE) et donnent naissance à un courant photo-électrique transitoire (fig. 1.1 et 1.2d).

Lors du balayage dans des sens opposés des paires électron-trou créées par un tir laser, deux phénomènes de collection de charges interviennent ; ils donnent sa forme au courant transitoire. En effet, le long de la trajectoire du tir, la distribution de charge créée va étirer le champ électrique selon cette même trajectoire et balayer en quelques picosecondes les charges se trouvant dans cette région. Ce balayage des charges dans des sens opposés provoque ainsi un pic de courant. C'est ce que l'on appelle le phénomène de "funneling". S'en suit alors un phénomène de diffusion beaucoup plus lent et ce, jusqu'à ce que la quasi totalité des charges soient collectées. Les charges restantes vont alors

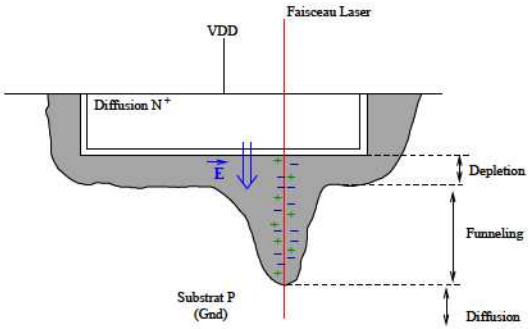


FIGURE 1.1: Effet d'un tir laser sur une jonction PN polarisée en inverse (N/substrat P).

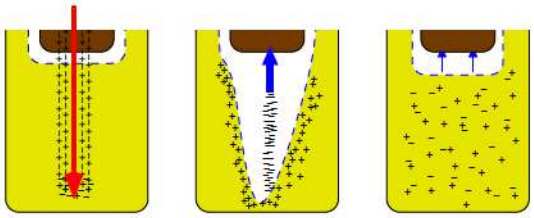
se recombiner sans provoquer d'effets particuliers sur le fonctionnement du circuit. La figure 1.2 représente les deux phénomènes intervenant dans la collection des charges après un tir laser ainsi que la forme d'onde du courant qui en résulte. Avec la forme d'onde du courant transitoire (figure 1.2d), on observe bien le pic de courant provoqué par le phénomène de "funneling" puis un niveau de courant plus faible mais présent un peu plus longtemps lors de la diffusion.

Dans les circuits CMOS, le champ électrique nécessaire à la création d'un photo-courant est localisé au niveau des jonctions PN formées entre les zones de diffusion des transistors (drain ou source) et le substrat (de type P pour un NMOS et de type N pour un PMOS). En effet, n'importe quelle jonction PN du circuit, avec un champ électrique assez fort résultant d'une polarisation inverse, peut permettre l'apparition d'un photo-courant.

Comme précisé dans la partie 1.2.1, un photo-courant n'entraîne pas forcément un transitoire de tension. La transformation d'un photo-courant en un SET est déterminée par l'état du circuit, c'est-à-dire l'état des transistors, mais aussi la localisation du tir sur le circuit. Selon l'état du circuit certaines zones sont plus sensibles que d'autres. Cette notion de sensibilité est expliquée dans la partie suivante.

1.2.4 Notion de zones sensibles et SET

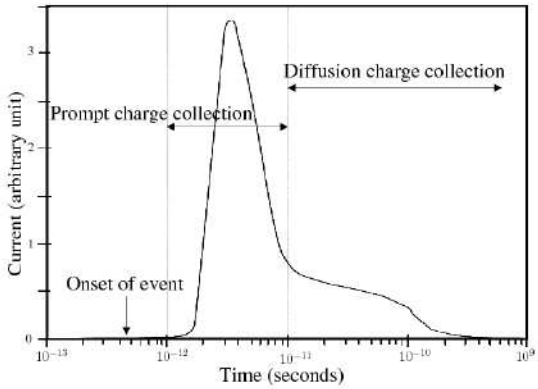
Le passage d'un photo-courant à un SET est déterminé par la position du tir laser sur le transistor ainsi que par l'état de ce dernier. Pour illustrer cela, prenons l'exemple d'un



(a) Création de charges pendant le tir laser

(b) Collection des charges dite "funneling"

(c) Diffusion des charges restantes

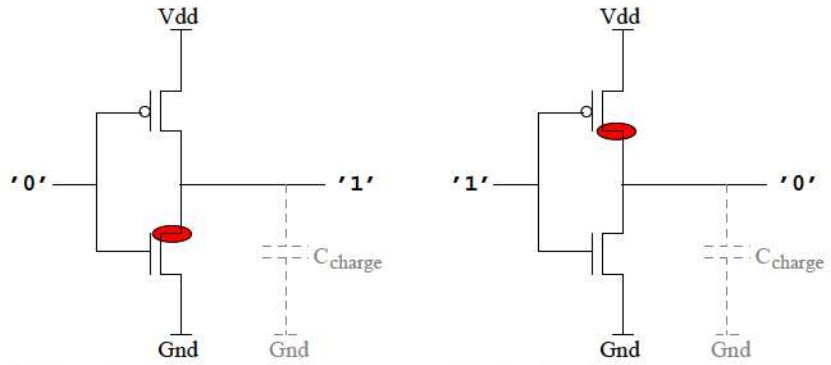


(d) Forme d'onde du courant transitoire [66]

FIGURE 1.2: Représentation des phénomènes à l'origine de la création d'un courant transitoire.

inverseur CMOS. La figure 3.16 montre le schéma au niveau transistor d'un inverseur pour les deux niveaux logiques possibles en entrée ainsi que ses zones sensibles, représentées par les zones colorées en rouge.

Dans la première configuration où l'entrée est à un état logique bas (fig. 1.3a), le transistor PMOS est passant et le transistor NMOS bloqué. Les quatre jonctions PN de l'inverseur (deux présentes sur les drains et deux sur les sources des deux transistors), si elles sont traversées par un faisceau laser, peuvent créer un photo-courant. Cependant seul un tir sur le drain du transistor NMOS provoquera un SET. En effet, le courant transitoire se propageant du drain (polarisé à VDD) vers le substrat (relié à la masse), a pour effet de décharger la capacité de sortie de l'inverseur (modélisant les portes logiques connectées en aval de l'inverseur). La sortie de l'inverseur passe alors temporairement



(a) Entrée de l'inverseur à l'état logique bas

(b) Entrée de l'inverseur à l'état logique haut

FIGURE 1.3: Schéma d'un inverseur CMOS avec localisation des zones sensibles.

au niveau logique '0'. Lorsque la zone sensible n'est plus affectée par le faisceau laser, le photo-courant cesse. L'inverseur reprend alors son état initial (niveau logique '1') grâce au courant fournit par l'alimentation de l'inverseur traversant le transistor PMOS qui est passant. Le tir laser a donc provoqué temporairement un changement de niveau logique en sortie de l'inverseur. Lorsque le tir laser a été stoppé, le niveau logique en sortie de l'inverseur est revenu à son état initial après le temps de charge de la capacité. Ce changement transitoire est bien un SET.

La création d'un photo-courant au niveau de la source du transistor NMOS n'aura aucun effet sur le niveau logique de sortie de l'inverseur ; le transistor NMOS est bloqué, la source est donc isolée de la sortie de l'inverseur. De même, pour la source et le drain du transistor PMOS, ces deux diffusions sont reliées au même potentiel que le puits N (VDD), donc le photo-courant créé ne perturbera pas le niveau logique de la sortie. Lorsque le niveau logique d'entrée de l'inverseur est à l'état bas, la seule zone sensible susceptible de permettre un transitoire de tension est le drain du transistor NMOS à l'état bloqué.

Par analogie de raisonnement, il est facile de montrer que lorsque l'entrée de l'inverseur est à un niveau logique haut (fig. 1.3b), la seule zone sensible susceptible de permettre un SET est le drain du transistor PMOS.

On peut donc généraliser en disant que les zones sensibles d'un inverseur CMOS correspondent aux drains des transistors bloqués. De plus, selon les données manipulées,

les zones sensibles ne sont pas les mêmes, il y a donc une dépendance de la localisation des zones sensibles par rapport aux données manipulées.

1.2.5 Single Event Upset

On a vu dans la partie précédente, au travers de l'exemple d'un inverseur CMOS, la transformation d'un transitoire de courant, créé par le tir laser et l'effet photo-électrique, en un transitoire de tension (changement de l'état de sortie de l'inverseur) lorsque le tir laser atteint une zone sensible. Cependant, pour parler de fautes, le transitoire de tension doit être pris en compte par un élément mémoire et ainsi affecter le calcul en cours pour générer une erreur sur le résultat final. On distingue alors deux mécanismes pour passer d'un SET à une faute, soit créer un SET directement dans un élément de mémorisation (registre, SRAM, etc.), soit créer un SET dans un élément combinatoire pour que celui-ci se propage jusqu'à un élément de mémorisation.

On s'intéresse ici à la création d'un SET directement dans un élément de mémorisation (registre, SRAM, etc.). Prenons l'exemple d'une cellule mémoire de type SRAM dont le schéma au niveau transistor est donné figure 1.4.

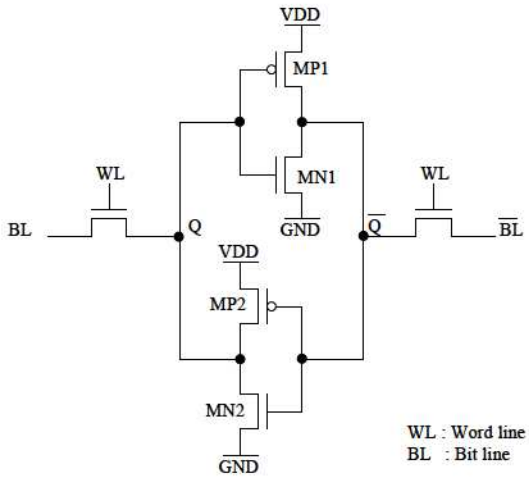


FIGURE 1.4: Schéma d'une cellule SRAM à 6 transistors.

Cette SRAM est constituée de six transistors. Quatre transistors constituent les deux inverseurs tête-bêches réalisant la fonction même de mémorisation, plus deux transistors permettant d'écrire ou de lire la valeur du bit mémorisé. Lors d'une écriture, la valeur

du bit à écrire est présentée sur *Bit line* (BL) et son inverse sur $\overline{\textit{Bit line}}$ ($\overline{BL}$). *Word line* (WL) permet de valider l'écriture. Pour la lecture, en mettant *Word line* à l'état haut, on peut alors lire la valeur mémorisée sur BL et la valeur mémorisée inverse sur $\overline{BL}$. Cette description des phases de lecture et d'écriture est simplifiée ; [60] apporte une explication plus détaillée de ces phases.

Pour la suite, on considère que l'état logique haut ('1') est mémorisé lorsque $Q = 1$ et $\overline{Q} = 0$. De même lorsque l'état bas ('0') est mémorisé, $Q = 0$ et $\overline{Q} = 1$.

Lorsque l'état logique haut est mémorisé, les zones sensibles sont donc les drains des transistors *MP1* et *MN2* (transistors bloqués des deux inverseurs constituant la SRAM). Si un tir laser intervient sur le drain de *MN2*, le niveau logique de la sortie de l'inverseur constituée des transistors *MP2* et *MN2* va donc passer d'un état logique haut à un état logique bas. Ce changement d'état va donc modifier l'entrée du deuxième inverseur constitué des transistors *MP1* et *MN1*. La cellule mémoire est alors placée dans un état instable. La sortie de ce second inverseur va donc passer d'un état logique bas à un état logique haut et la cellule mémoire retrouver un état stable. Ce changement d'état de la cellule mémoire est illustré avec la figure 1.5 où sont représentés les niveaux de tension de Q et $\overline{Q}$. Lorsque le tir laser intervient à 150 ns, le niveau de tension de Q décroit (état instable) jusqu'à faire changer le niveau de tension de $\overline{Q}$ (état stable).

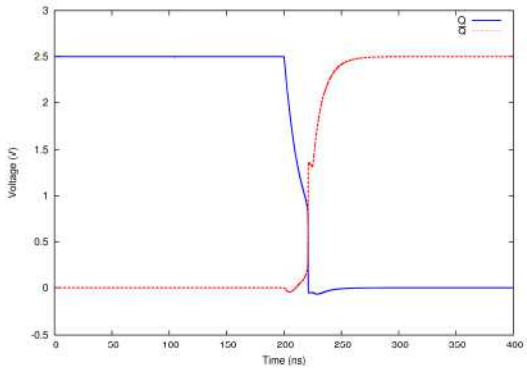


FIGURE 1.5: Basculement de la cellule mémoire lors d'un tir laser.

Lorsque le tir laser cesse, la cellule mémoire étant dans un état logique stable et bas, elle ne retrouve pas son état logique haut initial. L'état logique mémorisé est donc passé d'un niveau logique haut à un niveau logique bas. Le transitoire de tension provoqué par

le tir laser a changé l'état mémorisé par la mémoire et a donc provoqué une faute. On appelle ce type d'effet d'un tir laser, changeant directement la valeur du bit mémorisé, un effet de type Single Event Upset (SEU).

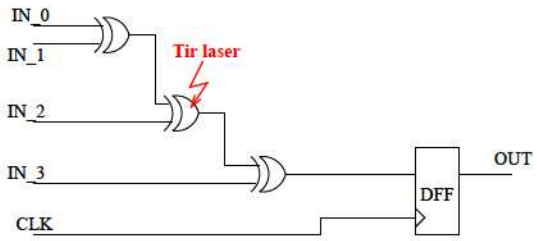
1.2.6 D'un SET vers une faute

Le second mécanisme pour obtenir une faute à partir de la création d'un SET concerne donc les transitoires de tension intervenant dans la logique combinatoire d'un circuit. Le SET créé va donc se propager à travers cette logique jusqu'à l'entrée d'un registre. Pour que le SET soit alors pris en compte par le registre, il faut que celui-ci soit présent lors de la capture de l'entrée du registre sur un front montant (ou descendant) de l'horloge du circuit intégré. Une contrainte temporelle sur l'instant d'injection est alors introduite.

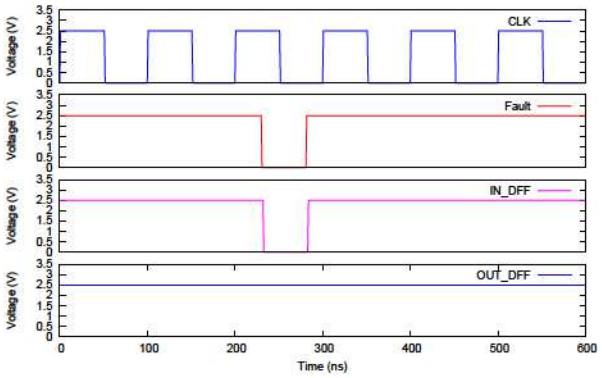
Pour mieux comprendre ce phénomène, prenons l'exemple de plusieurs portes XOR en cascades suivies d'un registre comme illustré par la figure 1.6a. Lorsqu'un tir laser atteint l'une des portes XOR et qu'un SET est créé, celui-ci va alors ce propager à travers les autres portes du circuit jusqu'à l'entrée du registre. Dans le premier cas, figure 1.6b, le SET atteint l'entrée du registre entre deux fronts montants de l'horloge et n'a donc aucun effet sur la sortie du registre. Le transitoire de tension n'occasionne aucune faute. Dans le deuxième cas, figure 1.6c, le transitoire de tension atteint l'entrée du registre lors de l'échantillonnage de celle-ci par l'horloge. À ce moment-là, le niveau logique bas provoqué par le transitoire de tension est mémorisé (à la place d'un niveau logique haut). Le reste du circuit est impacté par cette mauvaise mémorisation, on a donc une faute.

1.2.7 Effet destructif (Latch-Up)

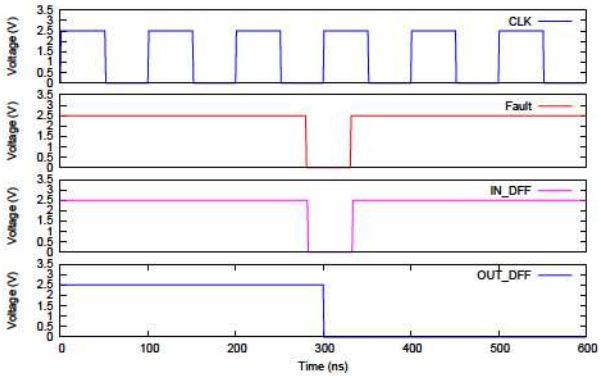
Ce phénomène apparaît lorsque la structure parasite thyristor (PNPN) du circuit CMOS est activée par le tir laser. Une fois activée, cette structure va créer un fort appel de courant entre les alimentations, ayant souvent pour résultat la destruction du circuit. La figure 1.7a représente la structure parasite dans la vue en coupe d'un inverseur CMOS, la figure 1.7b représente le schéma électrique avec les différents éléments qui constituent la structure thyristor.



(a) Schéma logique de porte XOR en cascade et d'un registre.



(b) SET ne provoquant pas de faute.

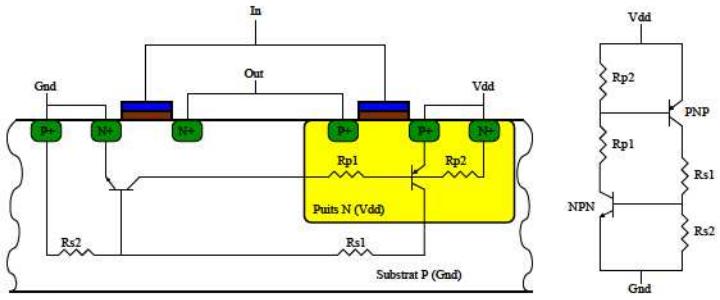


(c) Passage d'un SET à une faute.

FIGURE 1.6: Création d'un SET à travers une porte XOR se propageant jusqu'à l'entrée d'un registre.

Le transitoire de courant créé par le tir laser à travers les jonctions PN peut activer la structure parasite. De plus, si le gain de ces transistors parasites est supérieur à 1, la structure va entrer dans un état de verrouillage et créer un fort appel de courant, destructif pour le circuit.

Les solutions pour éviter ce phénomène peuvent être portées selon deux axes différents. Le premier concerne le design du circuit. En effet, une des solutions possibles est de réaliser le circuit en technologie *SOI* (Silicon on Insulator) dont l'effet sera d'éliminer la structure thyristor parasite. Une autre solution est de réduire le gain des transistors parasites pour qu'il soit inférieur à 1. Le deuxième axe d'action concerne cette fois-ci les paramètres du tir laser ; en limitant l'énergie du tir, on limite du même coup l'amplitude des impulsions de courant et donc l'amorçage de la structure parasite.



(a) Structure Thyristor dans la vue en coupe de l'inverseur

(b) Schéma électrique de la structure Thyristor

FIGURE 1.7: Structure thyristor d'un inverseur CMOS.

1.2.8 Bilan du mécanisme d'injection

Dans la description des différents phénomènes mis en jeu par un tir laser pour injecter une faute dans un circuit intégré, on a pu mettre en évidence plusieurs contraintes.

Premièrement, comme on a pu l'exposer dans la partie 1.2.6, une dépendance temporelle existe. Cette dépendance intervient lorsque l'injection de fautes est réalisée à travers les parties de logique combinatoire du circuit. Si le transitoire de tension se propageant à travers la logique combinatoire n'atteint pas un élément de mémorisation dans sa fenêtre de capture, aucune faute ne sera alors injectée. La synchronisation du tir laser avec le fonctionnement du circuit est alors très importante pour pouvoir obtenir des résultats exploitables.

Deuxièmement, une dépendance de l'injection de fautes aux données, liée à une dépendance spatiale de la position de tir ont été mise en évidence. Ceci s'explique par la dépendance aux données des zones sensibles. Selon les données manipulées, les zones sensibles ne seront par localisées sur les mêmes transistors constituant les différentes parties du circuit. Pour une même position de tir, la zone visée peut ne plus être sensible en fonction de la nouvelle donnée manipulée.

Cette dépendance aux données oblige à définir des modèles de fautes pour pouvoir connaître précisément les types de fautes injectées et ainsi avoir une meilleur analyse des injections réalisées sur le circuit intégré.

1.3 Modèles de fautes

Le plus souvent dans la littérature [43, 65, 35, 45, 46, 15], les fautes sont définies selon trois modèles bien connus. Les fautes de types *Bit-set**Bit-reset* et les fautes de types *Bit-flip*. Le troisième modèle, *Stuck-at*, est moins utilisé que les deux précédents, mais il est important de le définir ici pour la compréhension du reste du document et éviter un amalgame entre fautes de type *Collage* (ou *Stuck-at*) et *Bit-set**Bit-reset*.

On notera b un bit non fauté avec $b \in \{0, 1\}$ et b' un bit fauté avec $b' \in \{0, 1\}$.

1.3.1 *Bit-set**Bit-reset*

On définit une faute de type *Bit-set* par l'équation 1.4 :

$$b \rightarrow b' = 1 \tag{1.4}$$

et une faute de type *Bit-reset* par l'équation 1.5 :

$$b \rightarrow b' = 0 \tag{1.5}$$

Lorsqu'un bit est fauté et que sa valeur initiale passe de '0' vers '1' (respectivement de '1' vers '0') on parle donc de *Bit-set* (respectivement *Bit-reset*). En revanche lorsque la valeur initiale du bit est déjà à '1' (respectivement '0'), aucune faute n'est injectée, ou tout du moins la valeur du bit reste à '1'. Ce comportement montre une forte dépendance aux données des fautes correspondant à ce modèle. L'injection ou non d'une faute dépend de la valeur initiale du bit. Si on considère la valeur du bit comme aléatoire durant l'injection, sur un nombre de tentatives d'injections donnée, seule la moitié réussiront dans le cas de fautes de type *Bit-set* ou *Bit-reset*. Cette dépendance aux données peut être une explication à des taux d'injection faibles.

De plus, ce type de fautes peut être assez dangereux lorsque celles-ci sont injectées dans un circuit cryptographique. En effet, cela permet de réaliser des attaques de type *Safe error* [8] [33] et ainsi de retrouver rapidement la valeur de la clef secrète sans avoir à réaliser des analyses mathématiques poussées sur les résultats fautés (*cf.* partie 3.3.6).

1.3.2 Bit-flip

Ce deuxième modèle de fautes est indépendant des données. Cela se traduit par l'écriture de l'équation 1.6 :

$$b \rightarrow b' = \overline{b} \tag{1.6}$$

Quelle que soit la valeur initiale du bit('0' ou '1'), celle-ci est inversée par l'injection de faute. La valeur du bit fauté prend le complément de sa valeur initiale.

L'avantage de ce type de fautes est que le taux d'injection augmente de 50% par rapport aux fautes *Bit-set* ou *Bit-reset*, dû à l'indépendance aux données. Lors d'une campagne d'injection de fautes sur un circuit cryptographique, la capacité à injecter ce type de fautes permettra de réduire le nombre d'injections, pour pouvoir ensuite mener une analyse mathématique des résultats fautés en vue de retrouver la clef secrète. En effet, on obtiendra plus rapidement un nombre suffisant de chiffrements fautés avec des fautes de type *Bit-flip* grâce à l'indépendance aux données par rapport à une campagne d'injection de fautes de type *Bit-set* ou *Bit-reset*.

1.3.3 Collage

Les fautes de type *Collage* (ou *Stuck-at*) concernent les fautes où la valeur du bit est collée à une valeur fixe ('0' ou '1'). Peu importe la valeur initiale du bit fauté, celle-ci est fixée à une valeur et même en présence d'une tentative de ré-écriture d'une valeur opposée, la valeur ne changera pas. Une remise à zero complète du circuit permet parfois d'éliminer ce collage. Ce type de modèle de fautes est utilisé en test.

1.4 Influence des paramètres du tir laser

La position du tir laser ainsi que l'état du circuit à l'instant du tir, définissant les zones sensibles de celui-ci, ne sont pas les seuls paramètres à prendre en compte pour l'injection d'une faute. Les paramètres liés au laser lui-même ont une influence sur l'injection de fautes, de même que les conditions expérimentales telles que la distance de tir ou le mode d'injection : face avant ou arrière du circuit. Il est important de connaître

les effets de chacun de ces paramètres pour pouvoir optimiser au mieux l'injection et ainsi mieux comprendre le comportement du circuit testé vis-à-vis de l'injection de fautes.

1.4.1 Choix de la longueur d'onde

Le choix de la longueur d'onde est très important pour la génération d'événements singuliers dans un circuit intégré. En effet, celle-ci va conditionner l'énergie des photons injectés dans le circuit et donc la génération ou non de porteurs de charge par le phénomène photo-électrique. Pour que le phénomène photo-électrique se produise, il faut que l'énergie soit supérieure à l'énergie de bande interdite du silicium (1,12 eV), ce qui limite la longueur d'onde à un maximum de 1,10 μm (*cf.* partie 1.2.2). Malgré cette limitation, le choix de longueur d'onde est encore assez vaste, que ce soit dans le visible, l'ultraviolet ou l'infrarouge. En revanche les caractéristiques du silicium ainsi que le choix d'effectuer les tirs laser par la face avant ou par la face arrière vont imposer des contraintes sur le choix de la longueur d'onde à utiliser.

Tir par la face arrière

Lorsque l'on tire au laser sur un circuit intégré par la face arrière, la longueur d'onde a une incidence sur la profondeur de pénétration du faisceau laser dans le silicium. La longueur d'onde doit donc être correctement choisie afin d'atteindre les couches sensibles du circuit avec le moins de pertes. La figure 1.8, extraite de [36], montre pour différentes valeurs de dopage l'évolution du coefficient d'absorption en fonction de la longueur d'onde.

En se reportant à la figure 1.8, A.H.Johnston [21] préconise alors une longueur d'onde de 1,06 μm. Cette longueur d'onde permet d'avoir un coefficient d'absorption minimum pour la plupart des valeurs de dopage. La profondeur de pénétration du faisceau laser atteint 700 μm, ce qui est largement suffisant pour atteindre les zones sensibles d'un circuit intégré dont l'épaisseur de substrat est généralement comprise entre 200 et 300 μm. Néanmoins, H.Johnston n'exclut pas de travailler avec des longueurs d'ondes plus faibles mais recommande de choisir une longueur d'onde selon un critère important : avoir une profondeur de pénétration comprise entre 10 μm et 100 μm.

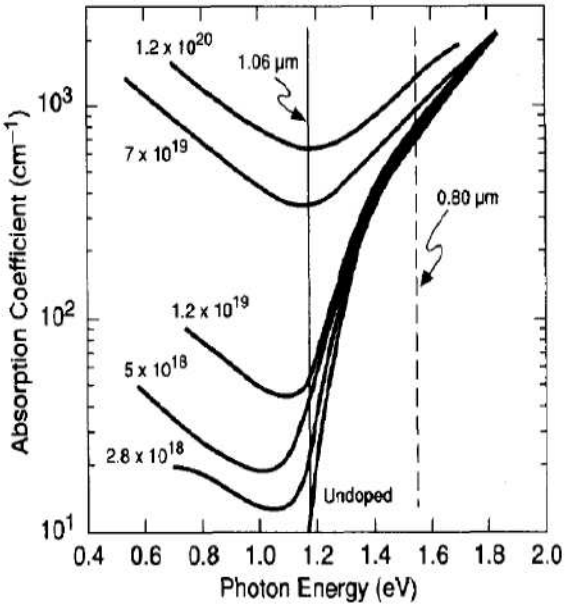


FIGURE 1.8: Coefficient d'absorption du silicium pour différentes valeurs de dopage en fonction de l'énergie du photon. [36]

Cependant, Melinger et al. [36] font remarquer qu'avec une longueur d'onde de 1,06 μm les écarts entre les coefficients d'absorption, pour différentes valeurs de dopage, sont assez grands, ce qui peut poser un certain nombre de problèmes pour obtenir les mêmes phénomènes sur différents circuits. Ils proposent donc une longueur d'onde de 0,80 μm, ce qui assure une profondeur de pénétration comprise entre 10 μm et 20 μm et réduit considérablement les écarts de coefficients d'absorption entre les différents dopages. Cependant, la préparation du circuit nécessaire pour obtenir une épaisseur de substrat de l'ordre de 20 μm est difficile et conduit souvent à la destruction du circuit. Cette longueur d'onde, malgré son efficacité, n'est peut être pas la plus facile à utiliser en pratique.

De plus, il faut garder à l'esprit qu'en traversant le substrat silicium du circuit cible, le faisceau laser va perdre de l'énergie. Au final lorsque celui-ci atteindra les zones sensibles, seulement une partie de l'énergie de consigne contribuera à l'effet photo-électrique au niveau de la zone sensible. En se basant sur l'article de Lewis et al. [32], pour un tir

laser en face arrière sur un circuit de 400 μm du substrat et une longueur d'onde de 1,1 μm, 66% de l'énergie est transmise à la zone sensible. Cependant, lorsque la valeur de dopage varie, le coefficient d'absortion varie d'un facteur 10 à 100. Cette forte variation ne facilite pas la répétabilité des tests d'injections entre différents circuits.

En prenant en compte ce paramètre, avec une longueur d'onde de 0,800 μm, permettant ainsi de minimiser au maximum les variations du coefficient d'absortion en fonction du dopage, l'énergie transmise devient quasi inexistante. Même en amincissant le substrat du circuit à 100 μm on atteint à peine les 0,01% d'énergie transmise. Une longueur d'onde plus appropriée permettant d'atteindre un pourcentage d'énergie transmise à la zone sensible acceptable est donc nécessaire.

La figure 1.9 permet d'avoir une idée du taux de génération des porteurs dans la zone sensible du circuit intégré (CI) en fonction de la longueur d'onde et différentes épaisseurs de substrat, ce pour une même concentration de porteurs égal à $5 \times 10^{18}\ cm^{-3}$. On remarque sur cette figure que le maximum est atteint pour une épaisseur de 50 μm et une longueur d'onde de 0,94 μm. A l'opposé, on trouve le minimum pour une épaisseur de 400 μm et une longueur d'onde de 1,04 μm. En gardant à l'esprit la figure 1.8 et les recommandations concernant le choix de la longueur d'onde faites précédemment, le meilleur compromis est d'avoir une épaisseur de substrat maximum de 100 μm et de travailler avec une longueur d'onde de 0,94 μm.

Tir par la face avant

Lorsque le tir laser est effectué par la face avant du circuit test, la longueur d'onde a une moins grande incidence sur l'apparition de SEE. En effet, le faisceau laser atteindra directement les zones sensibles, sans avoir à traverser au préalable le substrat. De même, le coefficient d'absorption, la profondeur de pénétration du faisceau ou encore le pourcentage d'énergie transmise aux zones sensibles ne sont plus des paramètres influençant l'apparition d'un SEE. Cette caractéristique permet une plus grande liberté dans le choix de la longueur d'onde. On pourra tout aussi bien réaliser des tirs avec une longueur d'onde dans le visible que dans l'infrarouge. Le choix sera majoritairement déterminé par le matériel à disposition.

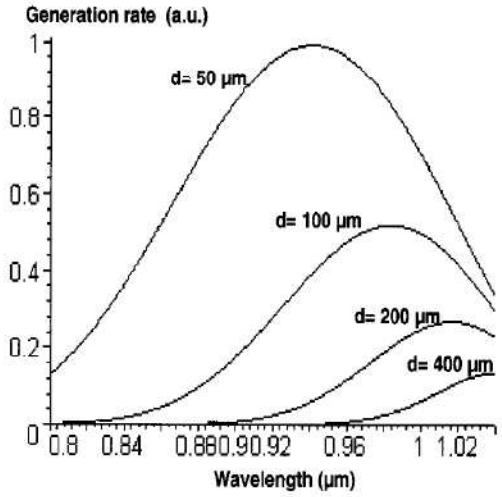


FIGURE 1.9: Taux de génération des porteurs en fonction de la longueur d'onde pour plusieurs épaisseurs de substrat. [32]

1.4.2 Tir par la face avant ou par la face arrière

Lorsque l'on veut effectuer un tir laser sur un circuit intégré, il peut se faire par la face avant ou par la face arrière du circuit. Les deux techniques ont leurs avantages et inconvénients. Il est important de les connaître pour ainsi faire le choix le plus adapté.

Pour un tir laser par la face arrière, le principal avantage est d'éviter les problèmes liés aux couches de métallisation. En effet, dans les circuits intégrés actuels, les couches de métallisation sont de plus en plus nombreuses et cela pose plusieurs problèmes pour un tir laser. Premièrement, le faisceau laser va être réfléchi par les différentes couches métalliques, si bien que seule une petite partie du faisceau va pouvoir atteindre les zones sensibles. Balasubramanian et al. [5] se sont intéressés dans leur étude à l'influence possible des différentes couches métalliques lors d'un tir laser sur l'apparition ou non d'une faute. Lors de test d'injections sur des chaînes d'inverseurs, il est apparu que plusieurs zones définies comme sensibles ne l'étaient finalement pas. Une analyse plus poussée du layout a montré que ces zones considérées comme sensibles étaient recouvertes par des pistes métalliques, ce qui les rendaient inaccessibles au faisceau laser par la face avant.

Deuxièmement, les circuits destinés à réaliser des opérations cryptographiques peuvent embarquer des protections contre l'injection de fautes par la face avant telles que

des *boucliers (ou shield)* réfléchissant intégralement le faisceau laser ou encore des détecteurs de lumière.

Néanmoins, une comparaison de tir par face avant et face arrière dans [32] montre que lorsque les couches métalliques sont peu nombreuses, les résultats obtenus sont identiques que l'on soit en face avant ou en face arrière. En revanche, Darracq et al. [10] ont été obligés d'adopter une approche par la face arrière pour pouvoir tester des mémoires SRAMs où les niveaux de métallisation étaient trop nombreux et perturbaient le tir laser par face avant. Le tir par la face arrière est plus efficace que le tir par la face avant lorsque le nombre de couches métalliques est supérieur à 4. Les circuits intégrés actuels possèdent en moyenne 6 couches métalliques, ce qui rend potentiellement le tir par la face arrière plus attractif.

Cependant, l'inconvénient majeur d'un tir laser par la face arrière est la préparation du circuit. En effet, le circuit doit être aminci pour avoir une transmission d'énergie du faisceau acceptable. Cette préparation est délicate puisque l'amincissement du circuit se fait le plus souvent de manière mécanique pouvant conduire à la destruction du circuit, or celui-ci doit conserver ces propriétés électriques une fois aminci pour que les résultats soit valides.

En règle générale, on privilégiera les tirs laser par la face arrière. Lorsque la face arrière du circuit n'est pas accessible ou que l'amincissement du substrat est impossible, on effectuera les tirs par la face avant. Néanmoins, aujourd'hui, avec l'augmentation croissante des niveaux de métallisation, les tirs par la face arrière sont privilégiés. On verra par la suite que certains avantages peuvent être tirés des tirs par la face avant et l'obligation du faisceau laser de traverser les couches de métallisation pour atteindre les zones sensibles.

1.4.3 Variation de la durée d'impulsion

Un changement de la durée d'impulsion du tir laser peut avoir plusieurs conséquences sur l'apparition d'un courant transitoire et donc d'une possible faute dans le fonctionnement du circuit intégré. Une des conséquences peut être par exemple d'augmenter significativement l'énergie nécessaire au tir laser pour le basculement de l'état d'une cellule mémoire SRAM [13]. Les figures 1.10(a) et 1.10(b) issues de [13] représentent re-

spectivement les comportements en courant et en tension obtenus par simulation d'un même transistor de type N soumis à un tir laser de même énergie mais avec des durées d'impulsions différentes ; à savoir respectivement 1 ns et 1 ps. On peut observer les pics des différents courants parcourant le transistor lors du tir laser ainsi que le changement des niveaux de tension, plus ou moins rapide, aux bornes de celui-ci pour les deux durée de tir.

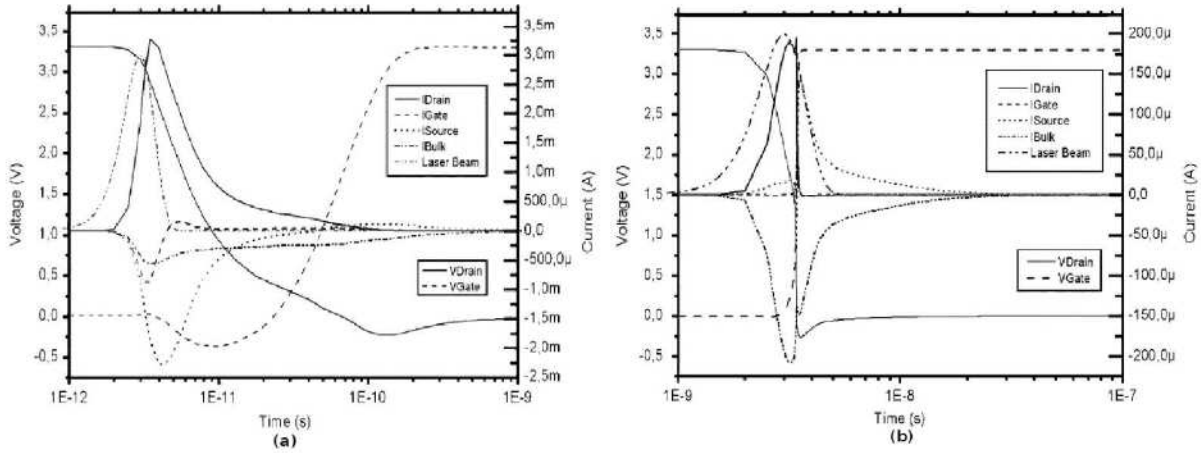


FIGURE 1.10: Réponse en courant et en tension d'un transistor de type N à un tir laser de 1 ps(a) et 1 ns(b). [13]

En comparant ces deux figures, on remarque que le niveau de la réponse en courant varie d'un facteur 100 d'une durée à l'autre. La durée d'impulsion a donc un effet sur la valeur du courant injecté à travers le transistor. Ce même transistor faisant partie d'un cellule SRAM, le courant injecté à travers celui-ci va conditionner ou non le basculement de cette cellule mémoire (*cf* partie 1.2). Une augmentation de la durée d'impulsion a donc pour conséquence d'augmenter l'énergie nécessaire pour opérer un changement d'état de la cellule SRAM. Plus précisément, pour une même énergie mais avec des durées d'impulsions différentes, les puissances vont être différentes, donc les taux de génération de porteurs vont l'être également. Ces taux, conditionnant la valeur du photo-courant généré, pour une même énergie mais une durée d'impulsion plus grande le photo-courant généré va être plus faible.

La Figure 1.11 montre l'évolution du niveau d'énergie du faisceau laser nécessaire pour faire changer d'état la cellule mémoire. Les durées peuvent être séparées en deux

catégories : les impulsions courtes (jusqu'à 100 ps) et les impulsions longues (au delà de 100 ps). Avec des impulsions courtes, la génération de charges est presque instantanée en comparaison du temps de changement d'état de la cellule et le phénomène de diffusion n'intervient presque pas dans la génération du photo-courant. En revanche, pour des impulsions longues, le phénomène de génération de charges est beaucoup plus lent en comparaison du temps de basculement de la cellule. Le phénomène de diffusion contribue majoritairement dans la génération du photo-courant, ce qui nécessite une énergie plus grande pour faire basculer la cellule.

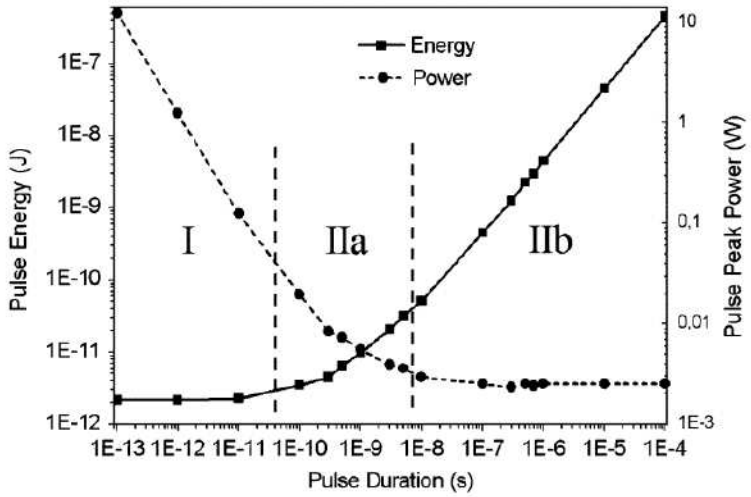


FIGURE 1.11: Niveau d'énergie laser minimal entraînant un SEU en fonction de la durée d'impulsion. [13]

1.4.4 Distance de tir

Lorsque l'on effectue un tir laser, que se soit avec une taille de spot large ou la plus petite possible, on souhaite transmettre le maximum d'énergie aux zones sensibles ciblées. Pour un tir par la face avant, on va donc simplement faire une focalisation du faisceau laser sur la zone sensible à atteindre en faisant abstraction des niveaux de métallisation.

En revanche, comme le montre la figure 1.12, si on garde la même démarche pour un tir par la face arrière, en effectuant le focus sur la surface arrière du circuit, lorsque

le faisceau laser va traverser le substrat, il va naturellement diverger, à cause de l'indice de réfraction du silicium beaucoup plus grand ($n = 3,96$) que celui de l'air. Au final, la taille de spot obtenue sur la zone sensible sera plus grande que celle d'un même tir par la face avant. La densité d'énergie sera donc plus faible, entraînant ainsi un photo-courant plus faible qui potentiellement ne suffira pas à provoquer une faute. Pour éviter ce problème, le focus doit être changé afin d'obtenir la même taille de spot atteignant la zone sensible, par la face arrière que par la face avant. Pour cela il suffit d'appliquer l'équation 1.7 donnée dans [10] :

$$z_1 = z_0 + \frac{e}{n}, \tag{1.7}$$

où z_1 est la nouvelle coordonnée de l'axe z afin d'obtenir le focus du faisceau laser sur la région active au travers de la face arrière, z_0 la coordonnée de l'axe z lorsque le focus est effectué sur la face arrière, e l'épaisseur du substrat et n l'indice de réfraction du silicium.

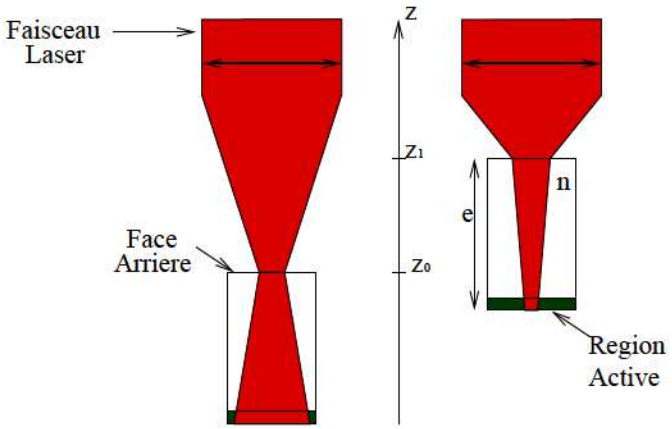


FIGURE 1.12: Focus du faisceau laser pour un tir par la face arrière. [10]

1.4.5 Taille du faisceau laser

Les mécanismes d'injections présentés dans la partie 1.2 ne prennent pas en compte la taille du faisceau laser par rapport à la technologie utilisée pour la fabrication du circuit intégré étudié. Pour une analyse théorique globale de l'effet d'un tir laser sur le circuit, cet oubli n'a pas de conséquence. En revanche pour une analyse plus précise et plus pertinente, il est indispensable de prendre en compte ce paramètre.

Comme le montre le tableau 1.2, la longueur de grille minimum des transistors ne cesse de décroître, pour atteindre aujourd'hui des tailles de l'ordre de 22 nm. Cependant, la taille minimale du faisceau laser est elle fixée par les lois optiques à 1 μm.

TABLE 1.2: Evolution des technologies silicium ces 20 dernières années.

Année	Taille de grille minimum
1990	1 μm
1995	0,35 μm
2000	0,13 μm
2005	65 nm
2010	32 nm
2013	22 nm

La figure 1.13 compare la taille de spot minimale d'un faisceau laser (1 μm) avec un transistor en technologie 1 μm (à gauche) puis avec un transistor en technologie 0.13 μm (à droite).

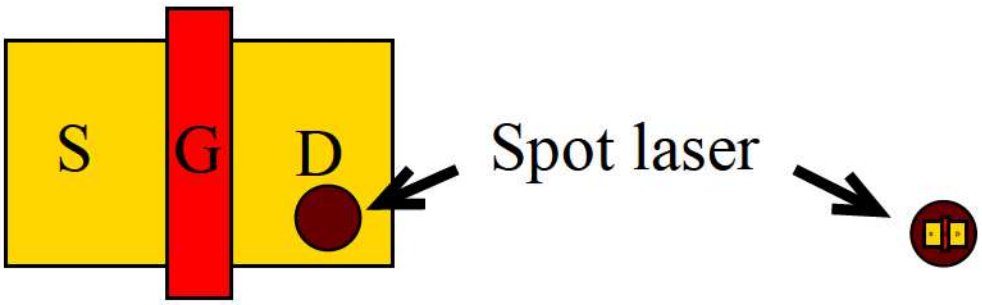


FIGURE 1.13: Comparaison entre deux transistors en technologie 1 μm (à gauche) et $0,13$ μm (à droite) et un spot laser de 1 μm de diamètre.

Avec les anciennes technologies, on voit bien que la taille de spot du laser ne pose pas de problème pour impacter seulement la zone sensible visée. Avec des technologies plus récentes, la taille de spot minimale est cette fois plus grande que la taille du transistor entier. Le spot laser englobe tout le transistor. Il devient alors compliqué, voire impossible, d'impacter seulement une zone sensible. De plus, la densité de transistors augmentant elle aussi au fur et à mesure que la longueur de grille minimum décroit, pour

une position fixe, le spot laser va impacter plusieurs transistors à la fois, voire même des portes logiques entières. De ce fait, plusieurs bits de données peuvent être impactés par le même spot laser.

La possibilité d'injecter des fautes de type *Bit-set* ou *Bit-reset* est alors remise en cause du fait de devoir seulement impacter une seule zone sensible. De même, le fait d'impacter plusieurs transistors avec le spot laser rend la possibilité de fauter un seul bit et non plusieurs bits ou plusieurs octets discutable.

Néanmoins, selon le besoin, il n'est pas forcement nécessaire d'avoir une taille spot permettant une injection de faute précise. Si on souhaite simplement injecter des fautes avec un coût réduit, sans contrôle sur les types de fautes injectées, une taille de spot large conviendra parfaitement. En revanche, pour un travail nécessitant le contrôle des types de fautes injectées, une taille de spot la plus petite possible est alors nécessaire.

Chapitre

2 Étude des modèles de fautes sur cellule SRAM

Préambule

Ce chapitre présente les modèles de fautes induits par un tir laser sur une cellule mémoire SRAM. *Des simulations électriques permettront de mieux comprendre la présence des modèles de fautes observés. Par la suite, des injections de fautes seront réalisées sur la mémoire* RAM *d'un micro-contrôleur. Le but de ces expérimentations est de confirmer les résultats obtenus sur une seule cellule mémoire. Ces travaux ont fait l'objet de plusieurs communications dans [53, 52, 58, 57].*

Contents

2.1 Introduction

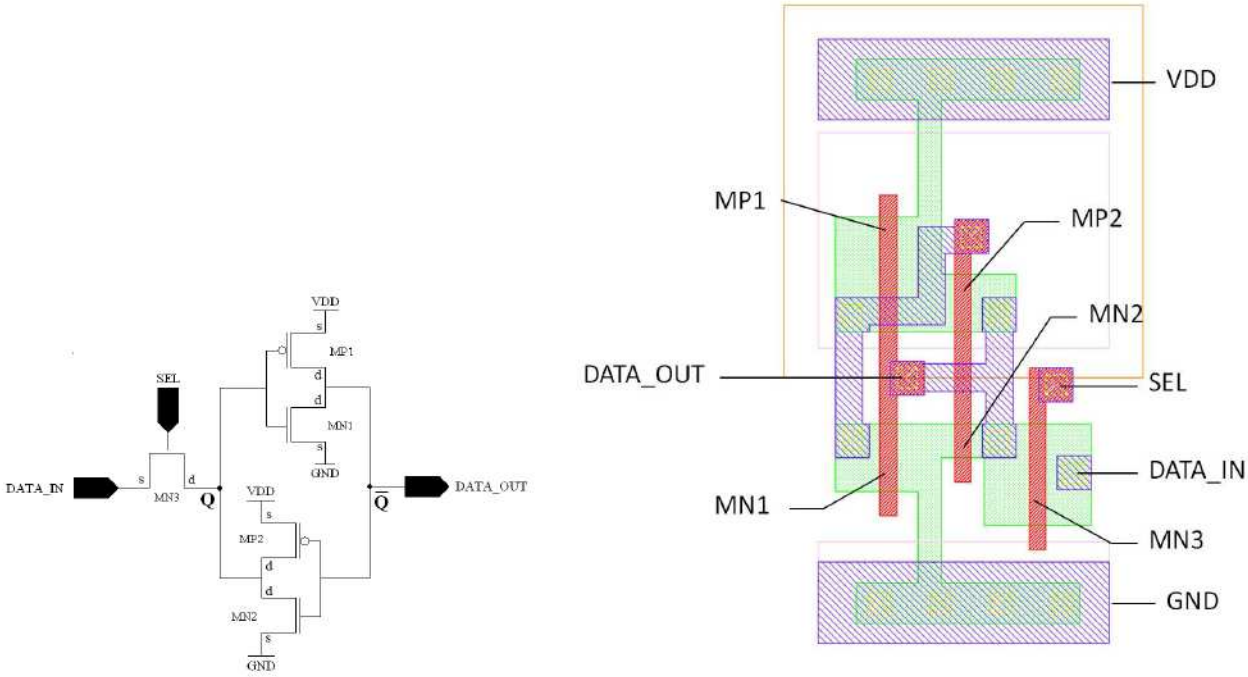
Les mémoires de type SRAM (*Static Random Acces Memory*) peuvent être utilisées dans les circuits cryptographiques pour stocker la clef secrète [1] ou différents calculs intermédiaires lors de l'exécution d'un algorithme de chiffrement. Ces mémoires constituent dès lors un point d'entrée pour l'injection de fautes et ainsi permettre une attaque visant à retrouver la clef secrète. Il est donc important de connaître les modèles de fautes possibles lors d'injections de fautes par tir laser. Cette connaissance des modèles de fautes possibles permettra d'anticiper les possibilités d'attaques de manière plus efficace ou encore d'améliorer les protections de ces mémoires contre les injections de fautes.

2.2 Le circuit SRAM

La cellule SRAM utilisée pour les différents tests des sections 2.2 à 2.7 est une SRAM de configuration (CSRAM) utilisée généralement dans les circuits à logique programmable pour mémoriser le *bitstream* nécessaire à la configuration du circuit et ainsi réaliser la/ou les fonctions souhaitées. Ce type de cellule mémoire utilise cinq transistors pour réaliser la mémorisation d'un bit, contrairement à une SRAM classique qui en utilise six (certaines variantes utilisent un nombre différent de transistors).

La figure 2.1 détaille le schéma au niveau transistor de cette cellule (figure 2.1a) ainsi que le layout de cette même cellule (figure 2.1b). Comme indiqué ci-dessus, la cellule est constituée de cinq transistors : deux inverseurs CMOS tête-bêches (*MP1*/*MN1* et *MP2*/*MN2*), réalisant la fonction de mémorisation du bit de configuration, et un transistor d'accès (*MN3*) permettant l'écriture du bit de configuration. La lecture du bit de configuration mémorisé s'effectue directement sur le nœud de sortie de l'inverseur constitué par les deux transistors *MN1* et *MP1*. Comme on peut l'observer sur la figure 2.1a, le bit lu est le complémentaire du bit mémorisé. Pour obtenir le bit mémorisé, il suffit d'ajouter un inverseur avant d'utiliser ce bit.

On peut noter que sur le layout de cette CSRAM, présenté à la figure 2.1b, les transistors PMOS *MP1* et *MP2* ont leur source en commun. De même, les transistors NMOS



(a) Schéma niveau transistor de la cellule CSRAM.

(b) Layout de la cellule CSRAM.

FIGURE 2.1: Schéma niveau transistor et layout de la cellule CSRAM.

MN1 et *MN2* ont leur source en commun. De plus, le transistor *MN2* a aussi son drain en commun avec celui du transistor d'accès *MN3*.

Pour plus de clarté, on considère que la cellule CSRAM est à l'état haut (ou "1") lorsque *DATA_OUT* est à l'état haut ($\overline{Q}$ = "1"). Par analogie, la cellule mémoire est considérée comme étant à l'état bas lorsque *DATA_OUT* est à l'état bas ($\overline{Q}$ = "0"). Pour mémoriser un bit, la valeur du bit est présenté sur l'entrée *DATA_ IN*. Lorsque le transistor d'accès *MN3* est passant (*SEL* = "1"), la valeur du bit mémorisé est actualisée avec la valeur présente sur l'entrée *DATA_IN*. Lorsque le transistor *MN3* est bloqué (*SEL* = "0"), la cellule mémoire est dans un mode statique : le bit de configuration est mémorisé.

Cette cellule mémoire CSRAM fait partie d'un circuit de test comportant plusieurs motifs d'éléments mémoires dans le but de tester leurs réactions aux injections de fautes par laser. Ce circuit a été réalisé en technologie CMOS 0,25 μm. Il utilise une tension

d'alimentation de 2,5 V. Une photo d'ensemble du circuit de test est donnée figure 2.2 avec un agrandissement de la partie du circuit où est implantée la cellule mémoire CSRAM. La CSRAM ne représente qu'une petite partie du circuit et a une surface de 9×4 μm^2. Pour faciliter les tests lors d'injections de fautes LASER par la face avant, la cellule CSRAM est dégagée de toute ligne de métal, évitant ainsi que le faisceau laser soit réfléchi et ne puisse atteindre les zones sensibles.

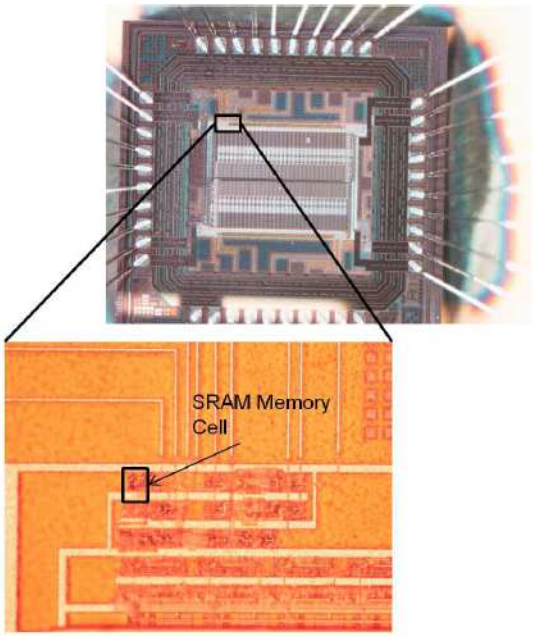


FIGURE 2.2: Vue d'ensemble du circuit de test et agrandissement de la zone contenant la CSRAM.

Pour la suite de l'étude, on préférera utiliser le terme SRAM pour désigner le cellule mémoire CSRAM. En effet, les résultats d'injection de fautes laser peuvent s'appliquer de manière équivalente aux cellules mémoires SRAM standards à six transistors.

2.3 Analyse théorique des zones sensibles

Comme présenté dans la partie 1.2.4, les zones sensibles aux injections de fautes par laser des circuits CMOS dépendent des données. Une première analyse du layout de la cellule SRAM a donc été réalisée pour estimer les zones où potentiellement des fautes

peuvent être injectées. La figure 2.3 présente le résultat de cette première analyse avec l'identification des zones sensibles sur le layout de la cellule SRAM.

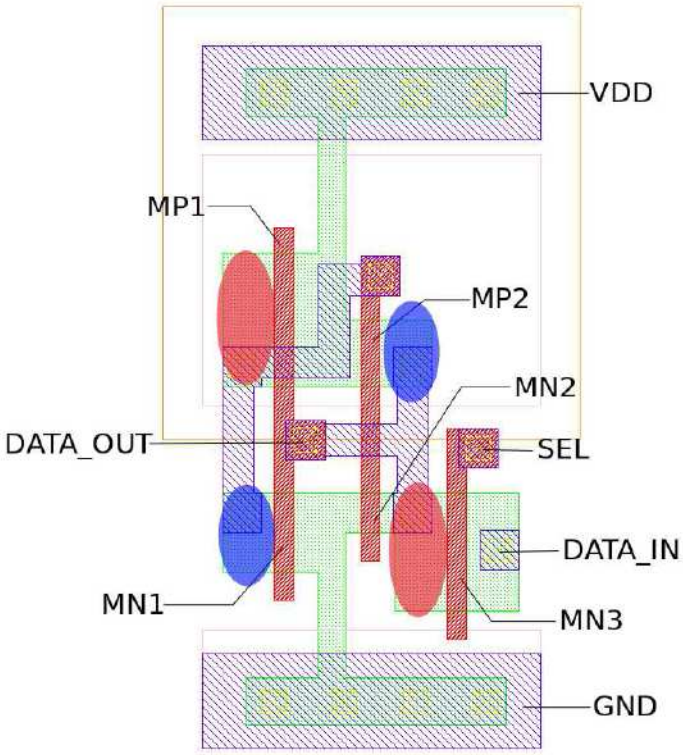


FIGURE 2.3: Layout de la CSRAM avec identification des zones sensibles : bleu pour l'état "1", rouge pour l'état "0".

Lorsque la cellule mémoire est dans un état haut, les transistors bloqués des deux inverseurs de la SRAM sont les transistors *MP2* et *MN1*. Conformément à la démonstration faite dans la partie 1.2.4, les drains de ces transistors bloqués sont considérés comme sensibles à l'injection de fautes par laser et sont donc identifiés sur le layout de la figure 2.3 en bleu.

De même, quand la mémoire SRAM est dans un état bas, les transistors *MP1* et *MN2* sont les transistors identifiés comme bloqués. Dans cet état de mémorisation, les drains de ces deux transistors (*MP1* et *MN2*) sont donc considérés comme sensibles à l'injection de fautes laser et identifiés sur la figure 2.3 en rouge.

Au final, on a donc quatre zones sensibles identifiées et dépendantes aux données : 2 zones sensibles dans l'état haut et 2 zones sensibles dans l'état bas.

Cette cartographie des zones sensibles de cette cellule mémoire est compatible avec le modèle de fautes de type *Bit-set* ou *Bit-reset*. En effet, si un tir laser atteint une des deux zones bleues lorsque la mémoire est à l'état haut, une faute peut apparaître et changer l'état mémorisé. L'état mémorisé passerait alors de l'état haut à l'état bas, ce qui correspondrait à une faute de type *Bit-reset*. À l'inverse, si le tir laser atteint une zone marquée en bleue à l'état bas, ces zones n'étant pas sensibles à l'état bas, l'état de la cellule mémoire ne peut être modifié. Respectivement, si un tir laser atteint une des deux zones rouges dans un état bas, l'injection d'une faute changerait l'état mémorisé et celui-ci passerait de l'état bas à l'état haut. Ce type de faute correspondrait au modèle de fautes *Bit-set*. De même que pour les zones marquées en bleues, si le tir laser atteint une zone rouge à l'état haut, l'état de la mémoire ne peut pas être modifié.

Cette première analyse des zones sensibles et par extension du type de fautes pouvant être injectées à l'aide d'un tir laser, exclut les fautes de type *Bit-flip*. Les quatre zones sensibles sont clairement distinctes et il n'existe pas de position où quel que soit l'état de la mémoire, celui-ci peut être changé par un tir laser. Une telle position signifierait que les zones sensibles se recouvrent les unes les autres.

Cependant on peut s'interroger sur le réalisme de cette analyse, celle-ci étant basée sur l'hypothèse qu'un tir laser ne peut toucher ou perturber qu'une seule zone sensible à la fois. Or, si on compare la surface de la cellule mémoire (4×9 μm^2) avec la taille minimum atteignable du spot laser (1 μm), on peut facilement envisager que le spot laser puisse atteindre, pour certaines postions de tir, plusieurs zones sensibles à la fois. De plus, la zone d'effet réelle d'un spot laser est toujours plus importante que sa taille. Une nouvelle analyse théorique des zones sensibles et des modèles de fautes possibles a donc été réalisée pour essayer de pendre en compte ces nouveaux paramètres. Le résultat est présenté sur la figure 2.4 où les zones sensibles sont identifiées sur le layout de la CSRAM.

On voit bien sur cette figure que les zones sensibles, associées aux modèles de fautes de type *Bit-set* (zones rouges) et *Bit-reset* (zones bleues), se recouvrent. Ces recouvrements laissent penser que les fautes de type *Bit-flip* sont réalisables avec un tir laser visant ces zones de recouvrement. En effet, à ces positions, un tir laser pourrait entrainer

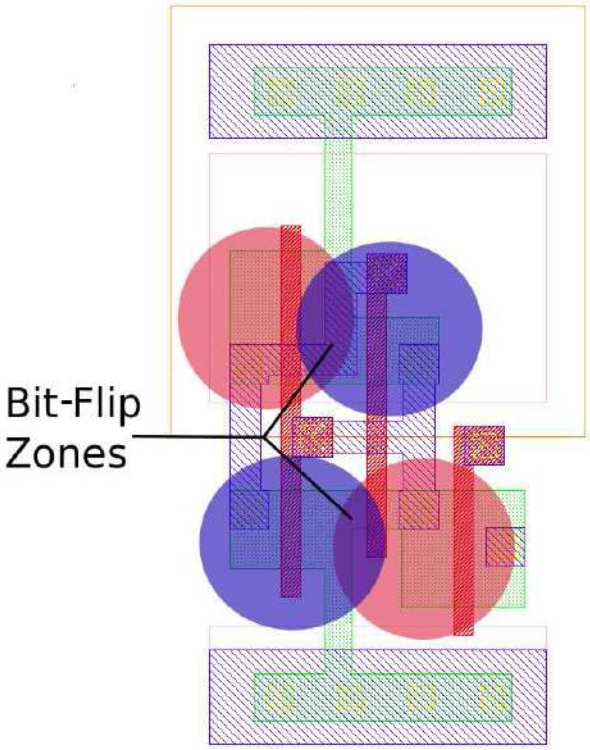


FIGURE 2.4: Layout de la CSRAM avec identification des zones sensibles incluant l'hypothèse de l'existence de zones de *Bit-flip*.

l'injection d'une faute quel que soit l'état initial de la cellule mémoire. Cela correspond bien au modèle de fautes de type *Bit-flip*.

La partie suivante s'attache à détailler les conditions expérimentales utilisées pour vérifier quels hypothèses et modèles de fautes sont réellement valides en pratique.

2.4 Conditions expérimentales

Dans cette partie, on commencera par détailler les particularités du banc laser utilisé pour les différentes injections de fautes, ainsi que les paramètres utilisés lors de nos expérimentations. Une seconde partie s'attachera à présenter la carte de test, liée à l'utilisation du circuit test contenant la cellule SRAM.

2.4.1 Description du banc laser

Le banc laser utilisé pour les différentes injections de fautes sur la cellule SRAM mais aussi sur le micro-contrôleur est un banc offrant le choix de plusieurs sources laser différentes mais ayant toutes une longueur d'onde dans l'infrarouge. Quatre sources sont disponibles : deux sources identiques de longueur d'onde 1064 nm permettant d'avoir des puissances en sortie de fibre optique allant jusqu'à 3 W et des largeurs de pulses comprises entre quelques nano-secondes et plusieurs secondes. Une troisième source de longueur d'onde 976 nm offre des puissances plus élevées avec la même plage de largeur de pulses. La quatrième source permet d'avoir une largeur de pulse fixe de 30 ps et une plage d'énergie en sortie de fibre optique allant jusqu'à 100 nJ. La longueur d'onde de cette source est fixée à 1030 nm. Le tableau 2.1 résume les différentes caractéristiques des différentes sources disponibles avec ce banc laser.

TABLE 2.1: Caractéristiques des différentes sources laser.

	Longueur d'onde	Durée d'impulsions	Puissance	Énergie
Source 1	1064 nm	5 ns $\rightarrow$ 1 s	0 $\rightarrow$ 3 W	-
Source 2	1064 nm	50 ns $\rightarrow$ 1 s	0 $\rightarrow$ 3 W	-
Source 3	976 nm	100 ns $\rightarrow$ 20 μs	0 $\rightarrow$ 25 W	-
Source 4	1030 nm	30 ps	-	0 $\rightarrow$ 100 nJ

Le faisceau laser choisi est ensuite focalisé sur le circuit grâce à un choix de lentilles optiques. Trois lentilles optiques disponibles permettent d'obtenir trois tailles de spot différentes : 1 μm (objectif x100), 5 μm (objectif x20) et 20 μm (objectif x5). Chaque optique atténue la puissance du faisceau laser transmis, de sorte que l'on obtient pour chacun des trois objectifs un coefficient de transmission. Ces coefficients sont résumés dans le tableau 2.2 :

TABLE 2.2: Table des coefficients de transmission des objectifs.

	Coefficient de transmission	Taille de spot
Objectif x100	26,5%	1 μm
Objectif x20	58%	5 μm
Objectif x5	67,5%	20 μm

Ces optiques sont fixées sur une table XYZ permettant de déplacer le faisceau laser sur la surface du circuit. Cette table offre une résolution de $0,1$ μm sur les trois axes. Le circuit test est quant à lui fixé sur une plaque de fixation pouvant être ajustée de manière manuelle sur les trois axes. Lors des campagnes de tests, le circuit est fixe. Ce sont les optiques qui bougent pour déplacer le faisceau laser. Cette disposition permet d'avoir des expériences fiables. En effet, le circuit est connecté à plusieurs câbles (l'alimentation, la communication, éventuellement certains signaux de contrôle), le bouger est difficile et risquerait de changer les conditions expérimentales. Enfin, un PC central permet de contrôler les différentes sources laser ainsi que les déplacements de la table XYZ.

La figure 2.5a montre le banc laser. Les différentes sources laser sont séparées du reste du banc contenant la table XYZ, les optiques de focalisation et la platine de fixation de circuit. Les différents faisceaux laser sont acheminés des sources vers le bâti via des fibres optiques. Une vue plus large de l'intérieur du bâti est présentée avec la figure 2.5b. On peut voir les optiques ainsi qu'un circuit de test fixé sur la platine.

(a) Vue d'ensemble du laser (bâti + sources laser). (b) Vue agrandie de l'intérieur du bâti.

FIGURE 2.5: Banc laser.

2.4.2 Carte d'interface

Pour pouvoir utiliser la cellule SRAM et plus généralement le circuit embarquant les différents motifs d'éléments mémoires, une carte d'interface est nécessaire. Cette carte permet de réaliser l'interface entre le PC de contrôle et le circuit test : transmettre les commandes d'écritures/lectures ainsi que de mettre en forme les données à écrire, transmises par le PC ou celles lues sur le circuit et pouvoir les transmettre de la carte vers le PC de contrôle. Cette interface est réalisée par un FPGA. La connexion avec le PC de contrôle est réalisée via une liaison série RS-232. Une photo de cette carte est donnée figure 2.6 où l'on peut aussi voir une carte d'adaptation pour le circuit. Elle permet de gérer les alimentations ainsi que l'adaptation des niveaux de tensions des signaux de contrôles et de données entre le FPGA (3,3 V) et le circuit test (2,5 V).

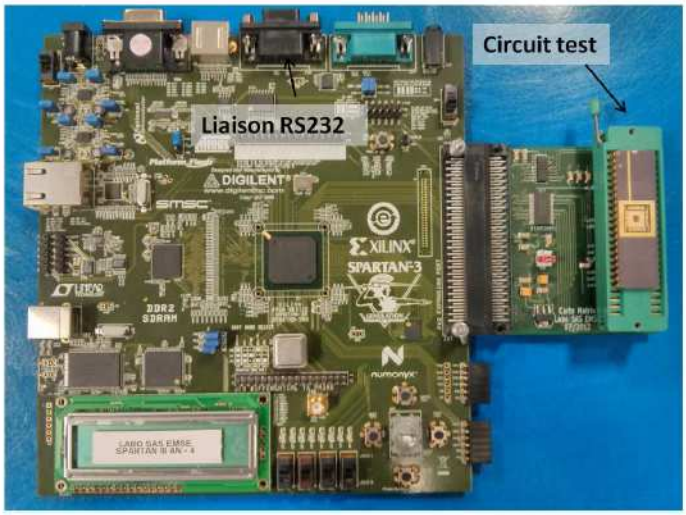


FIGURE 2.6: Carte d'interface pour le circuit test.

2.5 Cartographie des zones de sensibilité

Les différents tir laser présentés dans cette section ont été réalisé à l'aide de la source 1 du banc laser (*c.f.* tableau 2.1). Pour réaliser les différentes cartographies en fonction de la puissance du pulse laser, une surface de 10×10 μm^2 autour de la cellule SRAM a été scannée par la face avant avec un pas de déplacement de $0,2$ μm. Ici, le tir par la face

avant ne va pas influer sur l'injection de fautes puisque les pistes métalliques évitent au maximum le passage au dessus de la cellule mémoire, les zones potentiellement sensibles sont alors clairement dégagées et accessibles au faisceau laser.

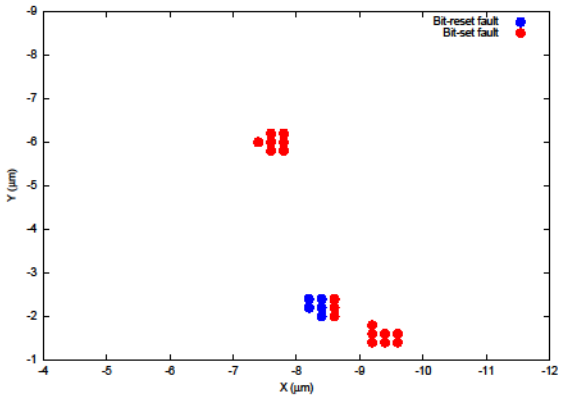
Pour chaque position, la mémoire est placée dans les deux états possibles de mémorisation (état haut et état bas). Une fois la valeur de la mémoire écrite et le transistor d'accès *MN3* bloqué, un tir laser est effectué, puis après une attente de quelques μs, l'état de la mémoire est lu. Si l'état lu après le tir laser est différent de l'état mémorisé, la position est ajoutée à la cartographie de sensibilité. Sur les différentes cartographies, on distingue les sensibilités aux fautes de type *Bit-set* (respectivement *Bit-reset*) pour un passage de l'état mémorisé bas à un état haut (respectivement d'un état haut à un état bas).

Les premières cartographies ont été réalisées pour des puissance, en sortie d'objectif, comprises entre $0,265$ W et $0,424$ W avec, à chaque fois, un spot laser de diamètre de 1 μm et une durée de pulse de 50 ns. Les fautes de type *Bit-set* sont identifiées en rouge et les fautes de type *Bit-reset* sont identifiées en bleu. Les figures 2.7a à 2.7f montrent l'évolution des zones sensibles en fonction de la puissance.

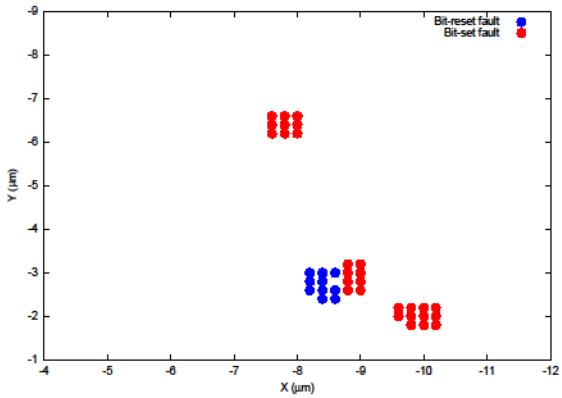
Les premières fautes apparaissent pour une puissance de 0,265 W. Pour des puissances au delà de 0,424 W, la cellule mémoire est irrémédiablement détruite. L'hypothèse de la cause de cette destruction est l'activation de la structure parasite thyristor (*cf.* partie 1.2.7) et donc d'un effet de latch up. En effet, à cette puissance aucun dommage structurel (coupure de piste métallique, gravure du silicium, etc.) sur la cellule mémoire n'a été relevé.

Au fur et à mesure que la puissance augmente, les zones sensibles s'étendent de plus en plus. Cette tendance confirme bien que la zone d'influence du spot laser est supérieure à 1 μm et que plus la puissance augmente, plus la zone d'influence est étendue. L'hypothèse faite dans la partie 2.3 sur des zones sensibles étendues par la zone d'effet du spot laser se vérifie donc bien avec ces cartographies, résultats des différentes injections de fautes laser réalisées sur la cellule SRAM.

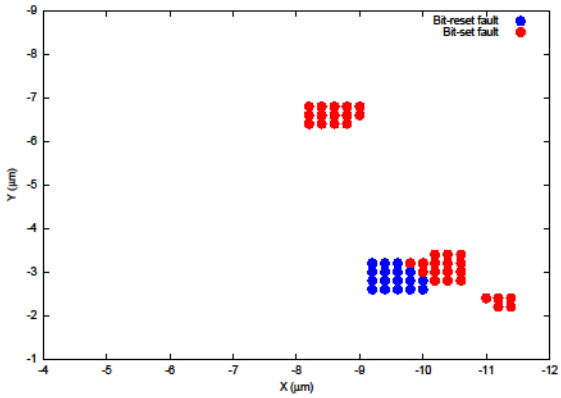
En analysant les six cartographies de la cellule SRAM pour une gamme de puissance allant de 0,265 W à 0,424 W, on peut identifier les deux zones correspondantes aux fautes de type *Bit-set*. Ces deux zones rouges correspondent aux drains des transistors



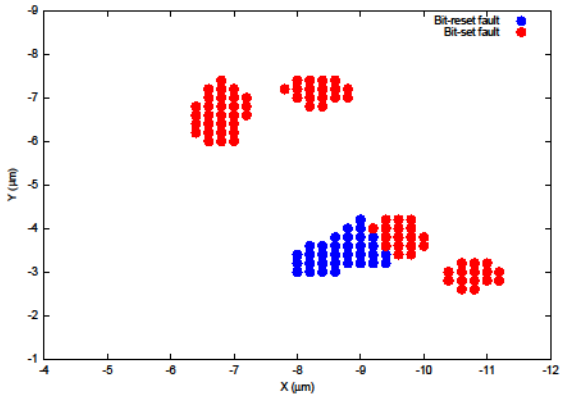
(a) Puissance = 0,265 W



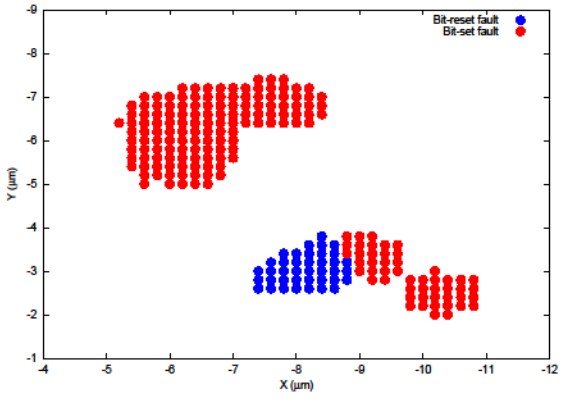
(b) Puissance = 0,290 W



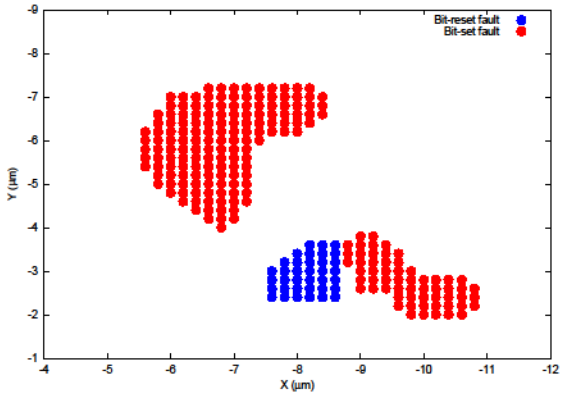
(c) Puissance = 0,320 W



(d) Puissance = 0,370 W



(e) Puissance = 0,400 W



(f) Puissance = 0,424 W

FIGURE 2.7: Cartographie des sensibilités pour des puissances entre 0,265 W et 0,424 W et un spot laser de 1 μm.

MP1 et *MN2/MN3*. En revanche, il n'y a qu'une zone correspondant aux fautes de type *Bit-reset* (zone bleue), le drain du transistor *MN1*. On peut supposer que le drain du transistor *MP2* n'est pas sensible à l'injection de faute. L'analyse de cette zone non sensible sera abordée plus loin. On peut aussi observer le détourage fait par les deux pistes métalliques de la cellule sur les deux zones de *Bit-set* sur les figures 2.7e et 2.7f. Ces deux zones contournent les pistes au fur et à mesure que les zones sensibles s'étendent dû à l'augmentation de la puissance du tir laser (*cf.* figure 2.1b).

La principale caractéristique observée avec ces différentes cartographies est l'absence de fautes de type *Bit-flip*. Quelle que soit la position du tir laser, les zones de *Bit-set* et de *Bit-reset* ne se recouvrent pas. Pour plusieurs positions, les zones sont adjacentes sans se recouvrir. Ces premières expérimentations démontrent la non pertinence du modèle de fautes de type *Bit-flip* sur cette cellule SRAM.

Cependant, la question de la taille du spot utilisé (1 μm) peut se poser. La taille de spot laser est-elle finalement assez grande pour pouvoir obtenir le recouvrement entre elles des zones de *Bit-set* et *Bit-reset* ? Pour répondre à cette question, les mêmes expérimentations ont été conduites, cette fois-ci avec une taille de spot de 5 μm et en conservant une durée de pulse de 50 ns. La figure 2.8 montre la cartographie des zones sensibles réalisée sur la cellule SRAM avec une puissance de 0,75 W et une taille de spot de 5 μm. La densité de puissance correspondante reste en deçà du seuil de destruction relevé précédemment.

Comme pour les cartographies avec un spot laser de 1 μm, avec un spot de 5 μm, les deux zones de *Bit-set* sont bien identifiables. De plus, comme pour les cartographies précédentes, il n'y a pas de recouvrements des zones de *Bit-set* et *Bit-reset*. Cette cartographie permet donc de confirmer que seuls les modèles de fautes de type *Bit-set* et *Bit-reset* sont pertinents.

Les cartographies obtenues, avec une taille de spot de 1 μm et de 5 μm, montrent que même si les zones de *Bit-set* et *Bit-reset* ne se recouvrent pas, elles sont très proches. Cette constatation peut laisser penser que l'injection de faute de type *Bit-flip* n'est pas totalement impossible.

Pour mieux comprendre l'absence de fautes de type *Bit-flip*, des simulations SPICE d'injection de fautes ont été menées.

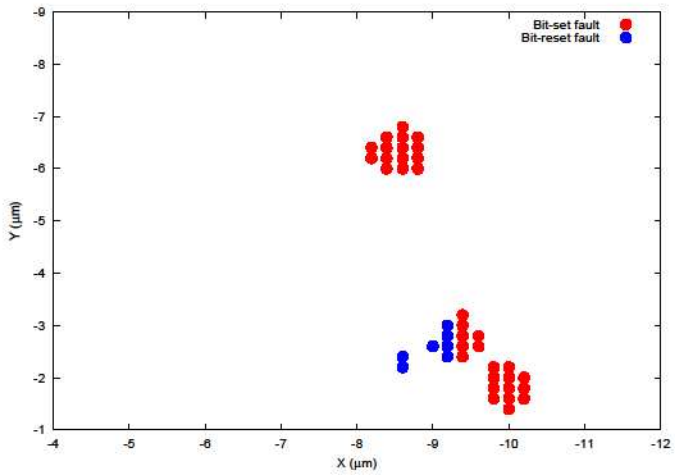


FIGURE 2.8: Cartographie des sensibilités pour une puissance de 0,75 W et un spot laser de 5 μm.

2.6 Simulation SPICE

2.6.1 Modèle de simulation

Les simulations présentées dans la suite de cette partie sont basées sur un modèle électrique de transistors MOS sous illumination laser présenté par Sarafianos et al. dans [56]. Dans ce modèle, le tir laser et plus particulièrement le photo-courant créé par celui-ci à travers les jonctions PN d'un circuit, est modélisé par une source de courant contrôlée en tension. Cette tension de contrôle est la tension présente aux bornes de la jonction PN considérée. L'amplitude du courant modélisant un tir laser est définie par l'équation 2.1 :

$$I_{laser} = (a * V + b) * \Omega_{laser} * S \tag{2.1}$$

La figure 2.9 montre un exemple de simulation d'un photo-courant créé par un tir laser au travers d'un jonction PN (courant transitoire definie en 1.2.3). La forme en double exponentielle du photo-courant simulé est la forme qui se rapproche le plus possible du photo-courant réellement créé par un tir laser. L'utilisation de cette forme en double exponentiel a été décrite et justifiée dans [4].

Dans l'équation 2.1, S est la surface de la zone sensible (en μm^2) où le courant est injecté, V est la tension aux bornes de la jonction PN polarisée en inverse, a et b sont

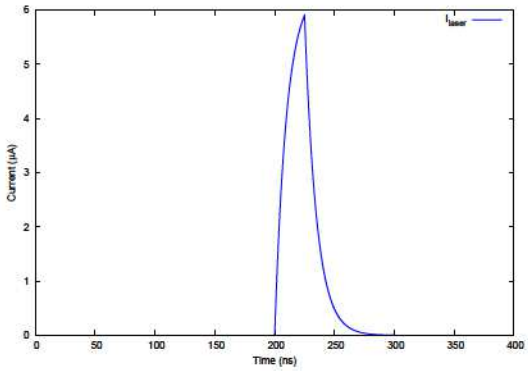


FIGURE 2.9: Forme d'un photo-courant simulé.

deux paramètres dépendants de la puissance du tir laser modélisé (en Watts) ainsi que de la technologie utilisée pour le circuit. a et b sont exprimés par les deux équations 2.2 et 2.3 :

$$a = p * P_{laser}^2 + q * P_{laser} + r \tag{2.2}$$

$$b = s * P_{laser} \tag{2.3}$$

P_{laser} est donc la puissance du tir laser que l'on veut simuler (en Watts), p, q, r et s sont des paramètres dépendant de la technologie du circuit cible. Dans [56], ces quatre paramètres ont été ajustés de manière à ce que l'amplitude du photo-courant simulé, pour différentes tensions d'inversion d'une jonction PN, corresponde aux mesures faites lors de tirs laser réels. Le modèle a été construit à partir de mesures faites sur des jonctions PN puis des transistors de type PMOS et de type NMOS.

Ω_{laser} est un paramètre utilisé pour prendre en compte l'affaiblissement du courant lorsque la distance entre la jonction PN et le spot laser augmente. Ce paramètre permet donc de prendre en compte la topologie du circuit. Selon la position du spot laser par rapport à la jonction PN considérée, l'amplitude du courant simulé (relatif au tir laser) ne sera pas la même. Ω_{laser} est calculé avec l'équation 2.4 :

$$\Omega_{laser} = \beta * \exp(-\frac{d^2}{c_1}) + \gamma * \exp(-\frac{d^2}{c_2}) \tag{2.4}$$

d est la distance entre la zone sensible (de la jonction PN considérée) et le spot laser (en μm), c_1 et c_2 représentent l'influence sur le spot laser, plus particulièrement sur

sa zone d'effet, des lentilles optiques utilisées pour concentrer le faisceau laser. β et γ permettent de prendre en compte l'influence de la durée du pulse laser.

Pour simuler un tir laser sur notre cellule SRAM, une source de courant, dont l'amplitude suit l'équation 2.1, a été connectée à chaque jonction PN du circuit. En effet, comme observé dans [56], même si les sources des transistors ainsi que les drains des transistors *ON* ne sont pas considérés comme sensibles, donc susceptibles de créer un photo-courant, lorsque un tir laser est effectué sur ces zones, celles-ci permettent tout de même de créer un photo-courant. Il est important de prendre en compte la génération de ces photo-courants dans la simulation car ils peuvent influencer l'apparition ou non d'une faute.

Conformément au layout de la cellule SRAM (figure 2.1b) et aux différentes diffusions partagées entre les transistors (les sources de *MP1/MP2* et *MN1/MN2* ainsi que le drain de *MN2/MN3*), sept sources de courant sont ajoutées au circuit représentant la cellule mémoire. Le schéma au niveau transistors final utilisé pour nos simulations est présenté dans la figure 2.10.

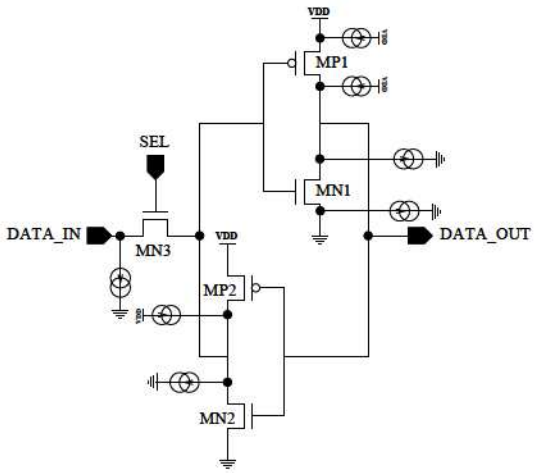


FIGURE 2.10: Schéma de la cellule SRAM avec les sources de courant modélisant le photo-courant induit par un tir laser.

Paramètres de simulation

Pour toutes les simulations, la surface de la SRAM a été quadrillée par pas de 0,5μm. Pour chaque position, les distances entre le spot laser et les différents drains et sources de la cellule mémoire sont calculées puis reportées dans les expressions I_{laser} des sources de courants correspondantes.
Pour notre circuit les paramètres de simulations du tableau 2.3 ont été utilisés :

TABLE 2.3: Table des coefficients de simulation.

	NMOS	*PMOS*
p	$4e^{-9}$	$9e^{-5}$
q	$-5e^{-7}$	$2e^{-4}$
r	$9e^{-6}$	$-5e^{-6}$
s	$4e^{-6}$	$1,2e^{-3}$

Pour le coefficient d'atténuation Ω_{laser} du spot laser, la durée de pulse utilisée étant de 50 ns, l'expression de ce coefficient est simplifiée dans l'équation 2.5.

$$\Omega_{laser} = \exp(-\frac{d^2}{c}) \tag{2.5}$$

Avec c valant $1,8\ \mu m^2$. Cette simplification ainsi que la valeur c ont été obtenues à partir de mesures expérimentales avec un pulse laser de 50 ns réalisées par A. Sarafianos dans le cadre de sa thèse.

2.6.2 Simulation des zones de sensibilité

La première simulation réalisée avec ce modèle électrique est la simulation de la cartographie des sensibilités de la cellule SRAM pour une puissance laser de 0,424 W. Le but de cette première simulation est double. Tout d'abord, valider le modèle de simulation. En effet celui-ci a été développé pour une technologie CMOS 90 nm, or notre cellule mémoire a été conçue en technologie CMOS 0,25 μm, il est donc important de valider la pertinence de ce modèle pour notre circuit avant de poursuivre l'étude avec des simulations plus avancées. L'autre but de cette simulation est de confirmer le comportement de la cellule mémoire et particulièrement l'absence d'une zone de sensibilité et de fautes de type *Bit-flip*.

Le résultat de cette simulation est présenté sur la cartographie des sensibilités de la figure 2.11.

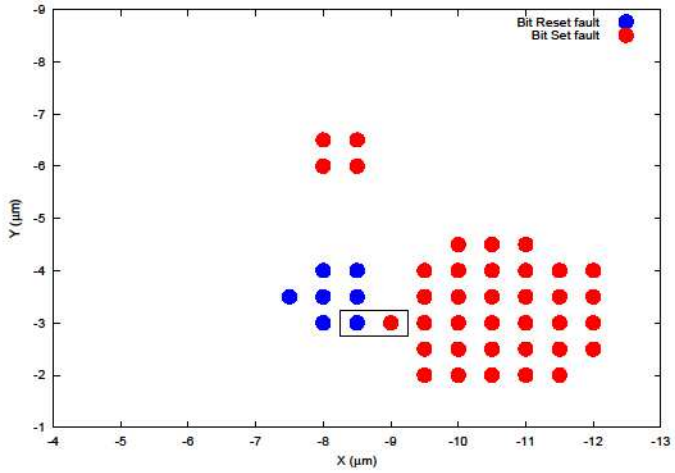


FIGURE 2.11: Simulation des zones de sensibilité de la SRAM.

Cette simulation des sensibilités est très semblable à la cartographie des sensibilités de la figure 2.7f, obtenue expérimentalement. Les zones de *Bit-set* en rouge sont bien présentes, de même que la seule zone de *Bit-reset* en bleu. La zone correspondant au drain du transistor *MP2* est bien manquante. De plus, il n'y a pas de zone de *Bit-flip* sur cette simulation. En effet, il n'y a pas de recouvrement des zones de *Bit-set* et *Bit-reset*. Le fait de retrouver les caractéristiques obtenues expérimentalement avec ce modèle de simulation confirme sa pertinence d'un point de vue qualitatif, même si les résultats obtenus ne peuvent pas être utilisés d'un point de vue quantitatif (le modèle a été développé pour une technologie 90 nm).

On peut donc utiliser ce modèle dans le but de mieux comprendre l'absence de fautes de type *Bit-flip*. Pour cela, les deux états possibles de la SRAM (état haut et état bas) sont simulés pour les deux positions mises en évidence sur la figure 2.11 par un rectangle. Ces deux positions représentent le seul point de contact entre la zone de *Bit-set* et la zone de *Bit-reset*.

2.6.3 Analyse des simulations sur l'absence de *Bit-flip*

Pour illustrer l'absence de fautes de type *Bit-flip*, une première simulation est réalisée à la position correspondant à la partie gauche du rectangle de la figure 2.11. Cette position est située dans la zone de *Bit-reset*. Pour la simulation, la SRAM est tout d'abord initialisé à l'état haut. La durée du pulse laser est fixée à 50 ns et la puissance à 0,424 W. Le tir effectif est simulé à partir de 200 ns sur les 400 ns de simulation.

Comme attendu, une faute de type *Bit-reset* est injectée (*DATA_OUT* passe de l'état haut à l'état bas à la suite du tir laser). L'injection de cette faute est illustrée dans la figure 2.12 où est représentée l'évolution des niveaux de tension des nœuds Q et *DATA_OUT* ($\overline{Q}$).

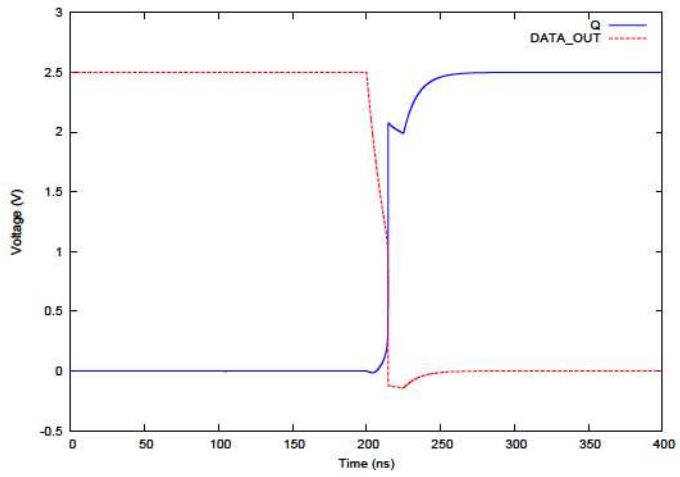
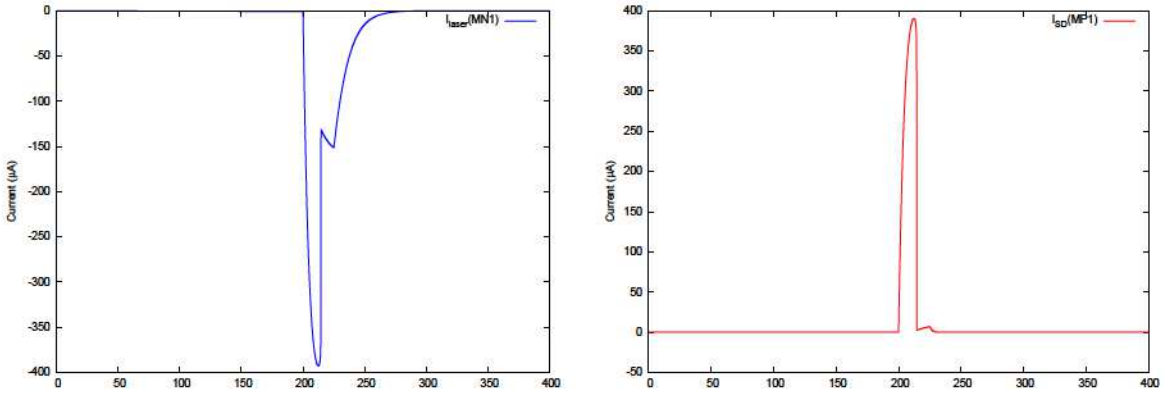


FIGURE 2.12: Simulation d'une faute *Bit-reset* : niveaux de tension des nœuds Q et *DATA_OUT*.

La position spatiale de tir est proche du drain du transistor *MN1*, identifiée comme sensible lorsque la SRAM est à l'état haut. Le photo-courant induit par le tir laser, modélisé ici par la source de courant I_{laser}, passe du drain de *MN1* vers le substrat de la cellule mémoire qui est relié à la masse (voir la figure 2.10). La circulation de ce courant à travers le transistor *MN1* a un effet de déchargement sur le nœud *DATA_OUT*. Cependant, le transistor *MP1* étant passant, un courant de contre balancement $I_{SD}(MP1)$ circule de *Vdd* vers le nœud *DATA_OUT* et a un effet de chargement sur ce nœud. Ces deux courants, I_{laser} et $I_{SD}(MP1)$, sont tracés sur les figures 2.13a et 2.13b.



(a) Photo-courant induit à travers le drain de *MN1*.

(b) Courant de contre balancement à travers *MP1*.

FIGURE 2.13: Simulation d'une faute *Bit-reset* : courants résultant du tir laser.

Le changement d'état de la SRAM est possible car le photo-courant induit par le tir laser (I_{laser}) est plus important que le courant de contre balancement ($I_{SD}(MP1)$) du transistor *MP1*. Cette différence est plus visible avec la charge électrique tirée du nœud *DATA_OUT* résultant des deux courants I_{laser} et $I_{SD}(MP1)$. La valeur absolue de la charge évacuée est tracée sur la figure 2.14.

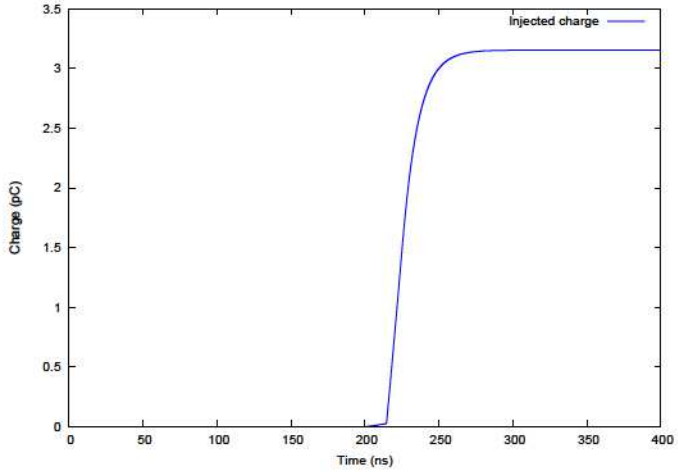


FIGURE 2.14: Simulation de la charge tirée du nœud *DATA_OUT* (valeur absolue).

Entre 200 ns et 220 ns, la charge du nœud *DATA_OUT* décroît (sa valeur absolue augmente) lentement car le photo-courant induit I_{laser} n'est supérieur au courant de

contre balancement $I_{SD}(MP1)$ que de seulement 10 μA. Durant cette phase, le niveau de tension de *DATA_OUT* est progressivement amené de son niveau initial de 2,5 V vers le niveau de tension de basculement de la SRAM ($\approx 1\ V$). Une fois le basculement de la SRAM effectué, la charge de *DATA_OUT* décroît rapidement. La cellule SRAM finit par se stabiliser à l'état bas. L'injection d'une faute de type *Bit-reset* est donc bien simulée.

Il est intéressant de noter qu'un second courant de contre balancement est induit par le tir laser. En effet, un photo-courant est induit à travers le drain du transistor *MN2* ($I_{laser}(MN2)$) tracé sur la figure 2.15 . Ce courant, circulant du nœud Q vers la masse, a pour effet d'essayer de maintenir ce nœud Q à un niveau de tension bas et donc d'empêcher le basculement de l'état de la cellule mémoire. Cependant ce photo-courant est trop faible pour éviter le basculement (de l'ordre de 40 μA avant le basculement de la cellule) et donc l'injection de la faute.

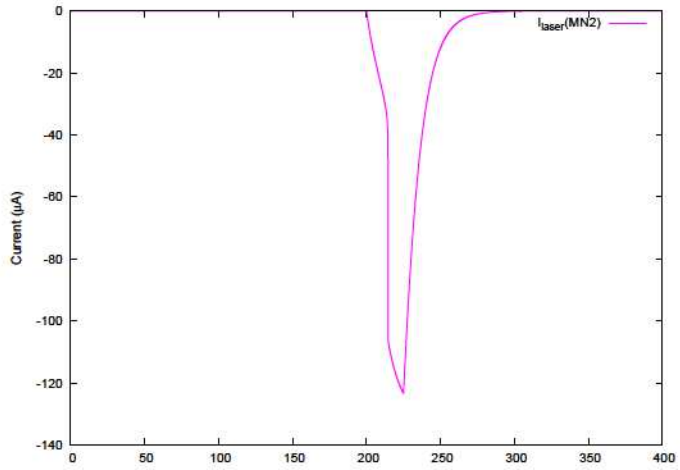


FIGURE 2.15: Simulation du photo-courant induit à travers le drain de *MN2*.

La deuxième simulation est effectuée à la même position mais avec cette fois la SRAM dans un état initial bas. Comme prévu, pour ces conditions d'injection, il n'y a pas de faute.

Les niveaux de tension des nœuds Q et *DATA_OUT* sont présentés figure 2.16.

À cette position de tir, la zone sensible la plus proche permettant une faute de type *Bit-set* est le drain du transistor *MN2*. Comme le montre la figure 2.16, le nœud Q subit une baisse de tension transitoire durant la simulation du tir laser (à partir de 200 ns

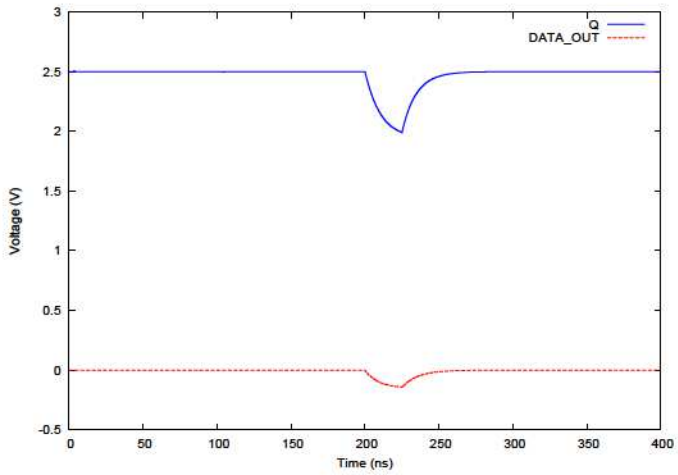


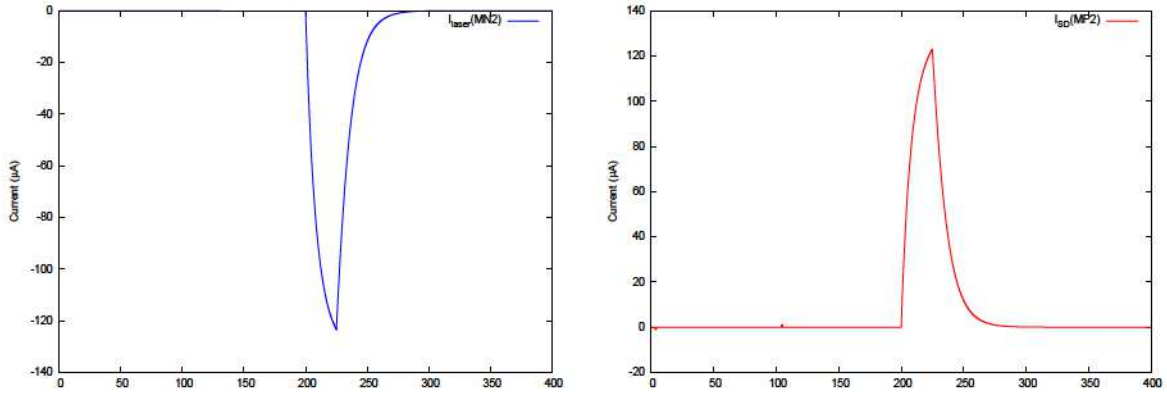
FIGURE 2.16: Simulation d'une tentative de faute *Bit-set* : niveaux de tension des nœuds Q et $DATA\_OUT$.

et jusqu'à 250 ns). Cette baisse de tension n'est pas assez importante pour atteindre le niveau de basculement de la cellule SRAM et ainsi changer son état de mémorisation. Si l'on s'intéresse au photo-courant induit par le tir laser sur le drain de *MN2* ($I_{laser}(MN2)$) ainsi qu'au courant de contre balancement du transistor *MP2* ($I_{SD}(MP2)$), tous deux tracés sur les figures 2.17a et 2.17b, on peut observer que le photo-courant est presque intégralement contre-balancé par le courant traversant *MP2*.

La figure 2.18 représentant la valeur absolue de la charge tirée sur le nœud Q, permet d'observer plus clairement cet effet de contre balancement empêchant l'injection de la faute.

Lors de la simulation du tir laser, le nœud Q subit une décharge de 0,02 pC, ce qui est relativement moins que la charge nécessaire de 0,03 pC injectée sur le nœud $DATA\_OUT$ pour induire un *Bit-reset* (voir figure 2.14 avant basculement de la cellule mémoire). Le photo-courant injecté par un tir laser à cette position (zone de *Bit-reset*), n'est pas assez élevée pour induire un *Bit-set*.

Ces deux simulations ont permis de comprendre pourquoi lorsque le tir laser est effectué sur la zone de *Bit-reset* (partie gauche) du rectangle de la figure 2.11 les fautes de type *Bit-set* sont impossible. Cette position est identifiée comme une des deux positions susceptibles de permettre l'injection de fautes de type *Bit-flip*. Les simulations suivantes



(a) Photo-courant induit a travers le drain de *MN2*.

(b) Courant de contre balancement à travers *MP2*.

FIGURE 2.17: Simulation d'une tentative de faute *Bit-set* : courants résultant du tir laser.

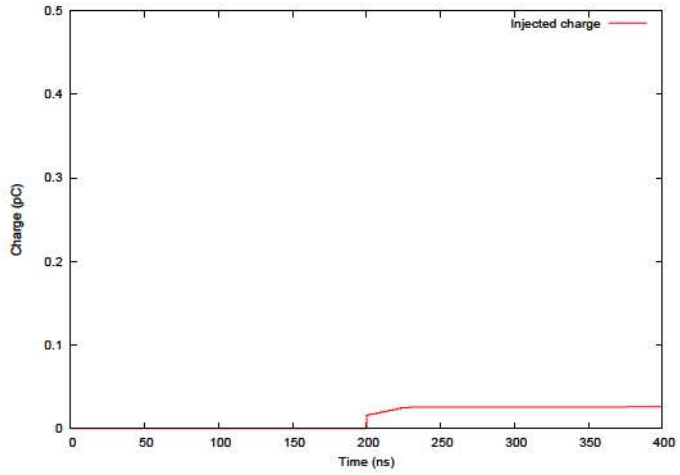


FIGURE 2.18: Simulation de la charge injectée sur le nœud Q.

ont pour but de vérifier ce résultat en simulant un tir laser sur la zone de *Bit-set* (partie droite) du rectangle de la figure 2.11. Les fautes de type *Bit-reset* ne devraient pas être possibles.

À cette position de tir, la première simulation est réalisée avec la cellule SRAM ayant un état initial bas. La figure 2.19 représente les niveaux de tension des nœuds Q et $DATA\_OUT$ pour cette simulation. Comme attendu, une faute de type *Bit-set* est in-

jectée, l'état de la cellule mémoire est changé (de l'état bas à l'état haut). Cette fois-ci, le photo-courant induit à travers *MN2* est assez fort pour ne pas être contre-balancé par le courant source-drain du transistor *MP2*, de la même manière que lors de la simulation d'une faute de type *Bit-reset*.

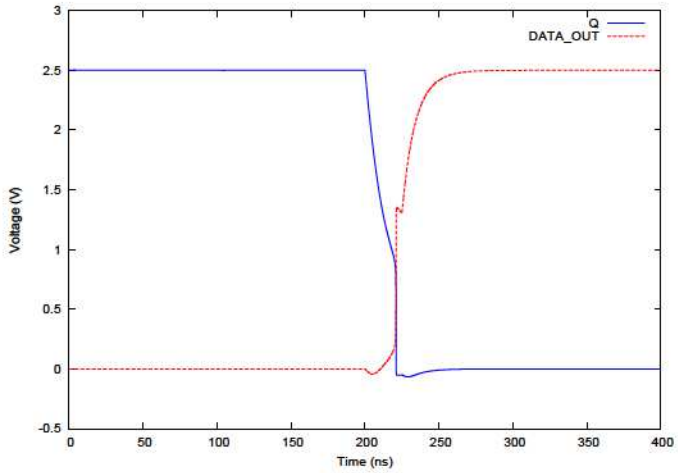


FIGURE 2.19: Simulation d'une faute *Bit-set* : niveaux de tension des nœuds Q et *DATA_OUT*.

Lorsque l'on simule la charge drainée sur le nœud Q (figure 2.20), on observe le même phénomène que lors de la simulation d'injection d'une faute *Bit-reset*. En effet, le photo-courant induit à travers *MN2* entraîne progressivement la tension du nœud Q de 2,5 V vers la tension critique de basculement de la SRAM ($\approx 1\ V$). Une fois ce niveau atteint, la charge électrique drainée (en valeur absolue) augmente rapidement jusqu'à ce que la SRAM soit stabilisée à l'état haut sachant que la charge drainée avant basculement est la même que pour une faute *Bit-reset* soit 0,03 pC.

Enfin, la dernière simulation concerne la tentative d'injection de *Bit-reset* dans la zone de *Bit-set* (partie droite du rectangle de la figure 2.11). La SRAM est dans l'état initial haut. On commence donc par tracer les simulations des niveaux de tension des nœuds Q et *DATA_OUT* sur la figure 2.21. Comme prévu, aucune faute n'est injectée.

De même que pour la seconde simulation, la charge électrique drainée sur le nœud *DATA_OUT*, figure 2.22, n'est que de 0,02 pC, ce qui est moindre par rapport au 0,03 pC nécessaires au basculement de la cellule SRAM lorsque celle-ci est à l'état initial bas. Ce

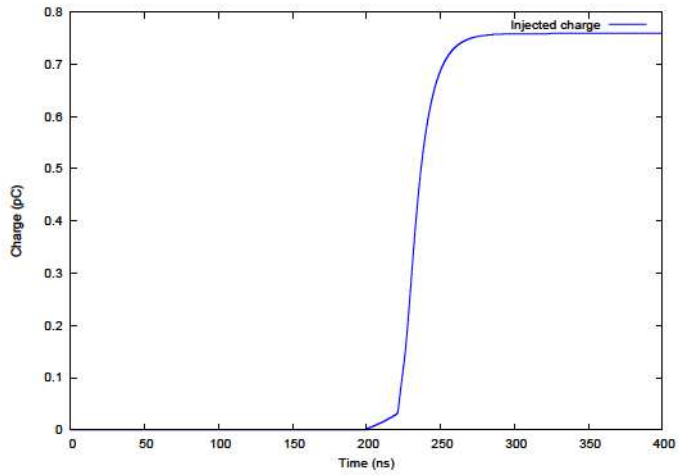


FIGURE 2.20: Simulation de la charge drainée sur le nœud Q.

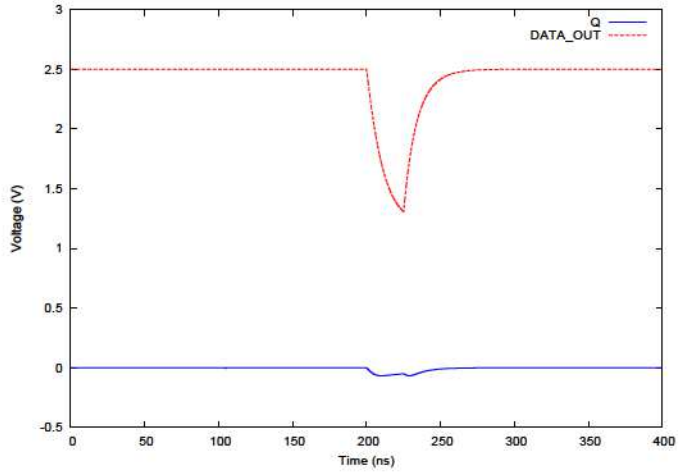


FIGURE 2.21: Simulation d'une tentative de faute *Bit-reset* : niveaux de tension des nœuds Q et $DATA\_OUT$.

faible niveau de charge électrique drainée sur le noeud $DATA\_OUT$ est la conséquence d'un photo-courant induit par le tir laser à travers *MN1* trop faible par rapport au courant de contre balancement du transistor *MP1*. Pour cette position, aucune faute de type *Bit-reset* ne peut être injectée.

Les deux positions considérées pour ces simulations étant identifiées comme les positions susceptibles de permettre l'injection de *bit-flip* et compte tenu des résultats des

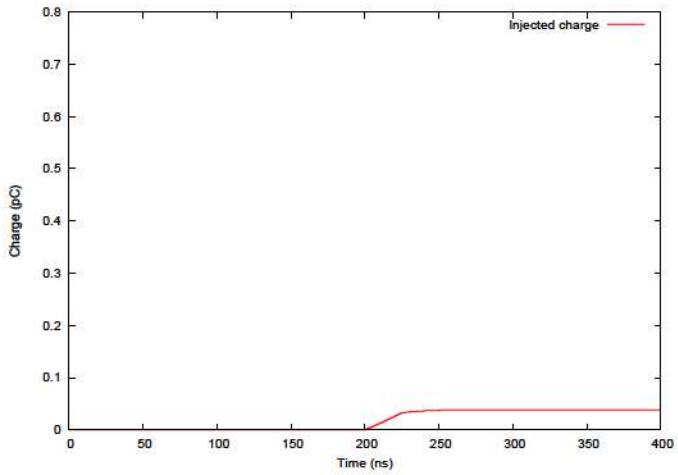


FIGURE 2.22: Simulation de la charge drainée sur le nœud *DATA_OUT*.

différentes simulations d'injection de fautes, on peut conclure qu'aucune faute de type *Bit-flip* ne peut être injectée sur cette cellule SRAM.

2.6.4 Analyse de la zone non-sensible

Comme évoqué dans la partie 2.5 lors des expérimentations ainsi que lors de la simulation des zones sensibles, la zone correspondant au drain du transistor *MP2* apparaît insensible aux tirs laser quelque soit la puissance utilisée.

Pour mieux comprendre l'absence de sensibilité de cette zone, une simulation de tir laser sur cette zone est réalisée. De même que lors de la simulation des zones sensibles, aucune faute n'est injectée. On peut voir sur la figure 2.23 représentant les niveaux de tension de Q et *DATA_OUT* que lors de la simulation du tir laser, le tension du nœud Q augmente sous l'effet du tir laser mais pas assez pour atteindre le seuil de basculement de la cellule SRAM. Lorsque le tir laser est terminé, les nœuds Q et *DATA_OUT* reprennent leurs niveaux de tension initiaux.

La figure 2.24 regroupe les différentes simulations du photo-courant injecté à travers le drain du transistor *MP2* (figure 2.24a), du courant de contre balancement du transistor *MN2* (figure 2.24b) ainsi que du photo-courant injecté à travers le drain des transistors *MN2/MN3* (figure 2.24c). On peut voir que le photo-courant du transistor *MP2* est légèrement supérieur ($0,3\ \mu A$) à la somme du courant de contre balancement

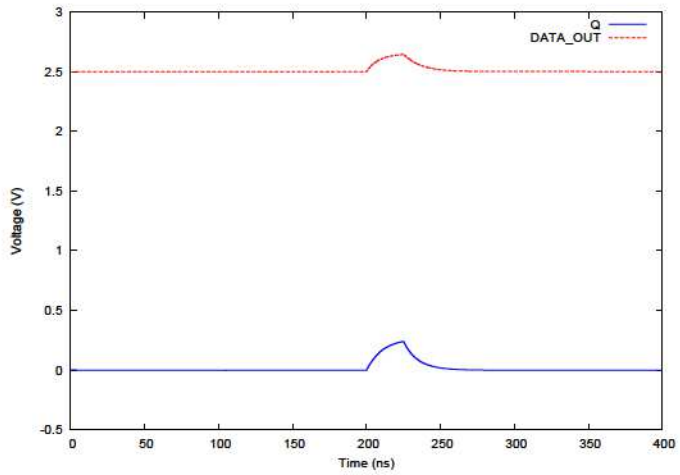


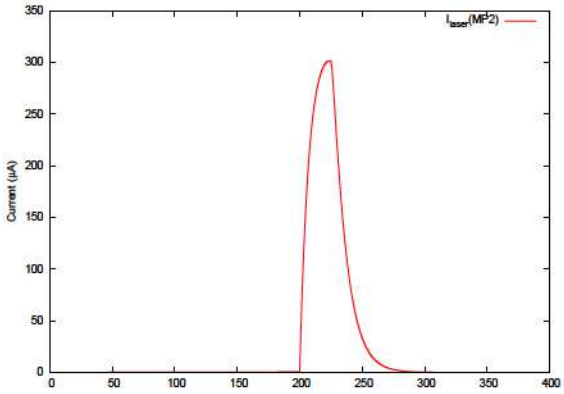
FIGURE 2.23: Simulation d'une tentative de faute *Bit-reset* : niveaux de tension des nœuds Q et *DATA_OUT*.

de *MN2* et du photo-courant injecté à travers le drain *MN2/MN3*. La combinaison de ces deux courants empêche le basculement de la cellule mémoire SRAM.

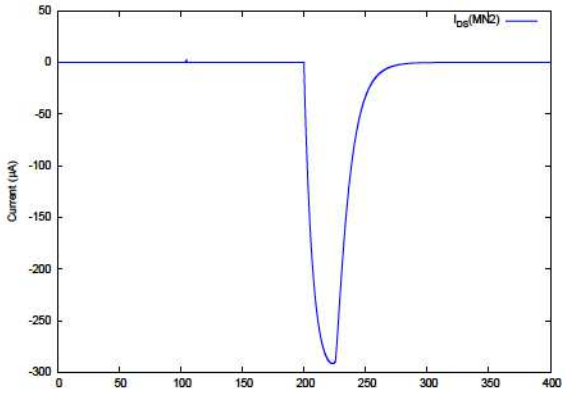
Cette simulation a permis de montrer que l'absence de sensibilité du drain du transistors *MP2* est due à un effet de compensation du photo-courant injecté. Cet effet de compensation est une combinaison du courant traversant le transistor passant *MN2* ainsi que du photo-courant injecté dans le drain de ce même transistor, partagé avec le transistor d'accès *MN3*. La présence de ce transistor d'accès offre un drain avec une surface beaucoup plus grande que celle du drain du transistor *MP2*. La surface du drain ayant une influence sur le photo-courant injecté lors d'un tir laser, cet écart de surface permet de générer un photo-courant à travers le drain de *MN2*, qui associé au courant de contre balancement de *MN2*, permet de contre balancer le photo-courant généré à travers le drain de *MP2*.

2.7 Tirs laser avec une source laser picosecondes

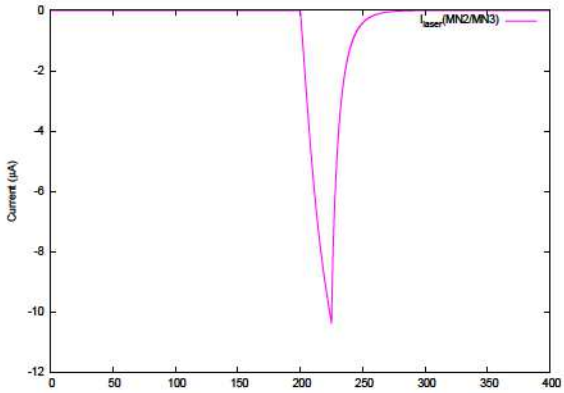
Des injections de fautes additionnelles ont été réalisées sur la cellule SRAM à l'aide de la source laser 4 (*c.f.* tableau 2.1) délivrant des pulses d'une durée fixe de 30 ps. Le



(a) Photo-courant induit à travers le drain de *MP2*.



(b) Courant de contre balancement à travers *MN2*.



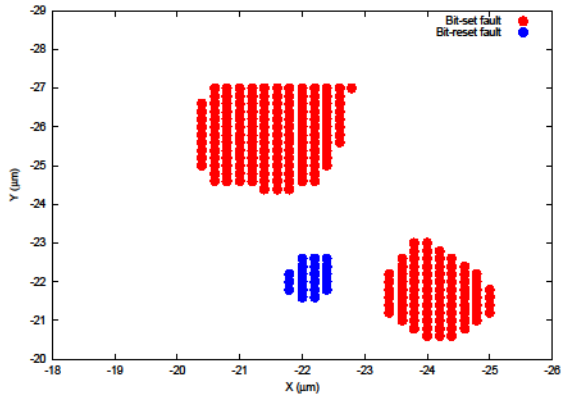
(c) Photo-courant induit à travers le drain de *MN2/MN3*.

FIGURE 2.24: Simulation d'une tentative de faute *Bit-reset* : courants résultant du tir laser.

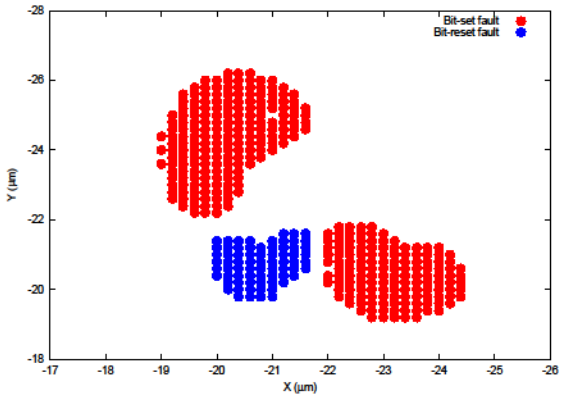
but de ces expérimentations était de confirmer l'absence de fautes de type *Bit-flip* mais aussi d'observer l'effet d'un pulse de cette durée sur les zones sensibles.

Pour les énergies de 2,65 nJ et 2,91 nJ en sortie d'objectif, on retrouve les mêmes zones sensibles que précédemment (partie 2.5), la drain de *MP2* est insensible et les deux zones de *Bit-set* et de *Bit-reset* ne se recouvrent pas. En revanche, pour une énergie de 3,18 nJ, les zones de *Bit-set* et de *Bit-reset* ne se recouvrent toujours pas, mais la deuxième zone de *Bit-reset* est sensible.

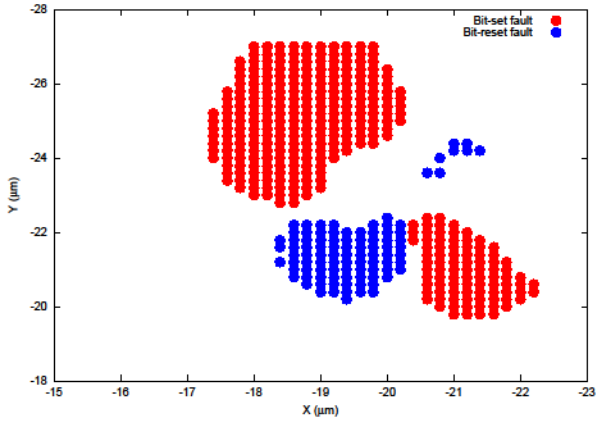
La sensibilité de cette zone au tir laser ayant une largeur de pulse de 30 ps peut



(a) Énergie du laser = 2,65 nJ



(b) Énergie du laser = 2,91 nJ



(c) Énergie du laser = 3,18 nJ

FIGURE 2.25: Cartographie des zones sensibles de la SRAM avec un pulse laser de 30 ps.

avoir plusieurs origines. Tout d'abord la densité d'énergie est plus forte que pour un pulse laser de 50 ns. L'effet de "funneling" est beaucoup plus important avec ce type de pulse, donc un photo-courant avec une amplitude plus grande est créé. L'effet de diffusion est plus court et n'est plus prépondérant sur l'effet de "funneling" [13]. Deuxièmement, la zone d'effet du spot laser devrait être très proche des 1 μm attendu. En effet, l'effet de "funneling" étant prépondérant, les charges n'ont pas le temps de se diffuser sur une zone beaucoup plus large que la taille du spot. Les effets de compensations dûs à la zone d'effet plus large (pulse de 50 ns) sont donc moindres et ne peuvent plus éviter le basculement de la cellule mémoire lorsque l'injection est effectuée sur le drain de *MP2*.

Cependant, cette zone reste relativement moins sensible que les autres comme le montre la figure 2.25c, où la zone sensible correspondant au transistor *MP2* est plus restreintes que les trois autres zones sensibles.

Des simulations, avec un modèle correspondant à des pulses laser d'une dizaine de picosecondes, pourraient permettre de vérifier ces hypothèses et mieux comprendre les effets de tels tirs sur une cellule SRAM.

2.8 Injection de fautes sur la mémoire RAM d'un micro-contrôleur

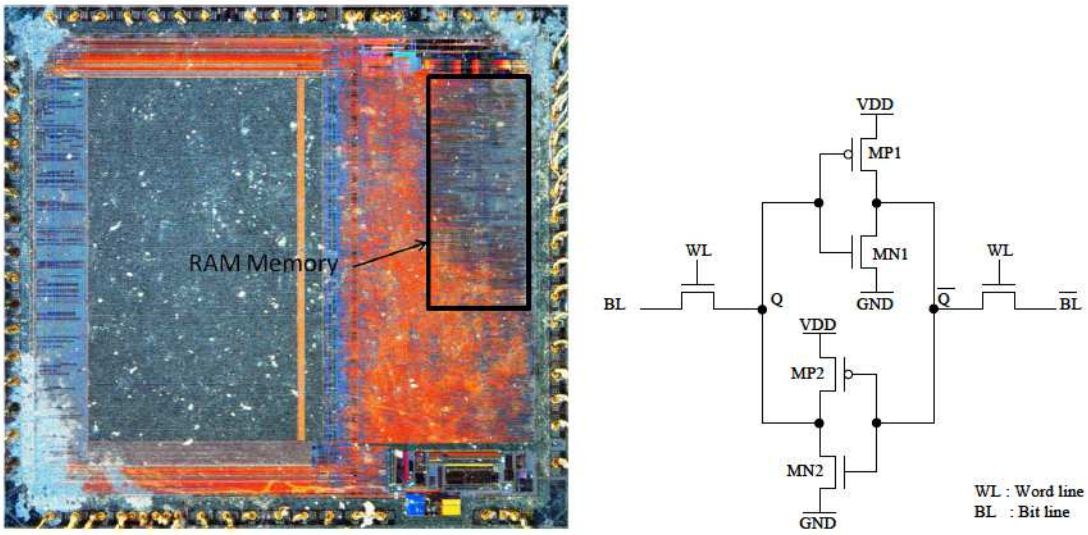
Pour confirmer les résultats obtenus sur une cellule mémoire isolée de type SRAM, des injections de fautes laser ont été conduites sur la mémoire RAM d'un micro-contrôleur comme décrit dans la partie suivante.

2.8.1 Description du circuit test

Le circuit utilisé est un micro-contrôleur 8-bits en technologie CMOS 0,35 μm présenté figure 2.26a avec une tension d'alimentation de 5 V. La mémoire RAM de ce circuit a une capacité de 4 ko. Cette mémoire est divisée en huit parties équivalentes, chaque partie contenant elle-même deux blocs de 256 octets. On peut considérer que chaque bit de cette mémoire RAM est constitué par six transistors : quatre transistors constituant les deux inverseurs tête-bêches plus deux transistors d'accès pour la lecture/écriture. La figure 2.26b rappelle le schéma d'une cellule SRAM à 6 transistors présenté dans la partie 1.2.5. A la vue de ce schéma ainsi que des hypothèse de zones sensibles exposées dans la partie 2.3 pour une SRAM à cinq transistors, chaque cellule SRAM de ce micro-contrôleur devrait présenter quatre zones sensibles (deux lorsque la cellule est à l'état haut et deux à l'état bas).

2.8.2 Conditions expérimentales

Pour les différentes injections de fautes, une carte d'adaptation est réalisée pour pouvoir communiquer avec le micro-contrôleur. La figure 2.27a illustre cette carte d'adap-



(a) Vue du micro-controleur avec sa mémoire RAM.

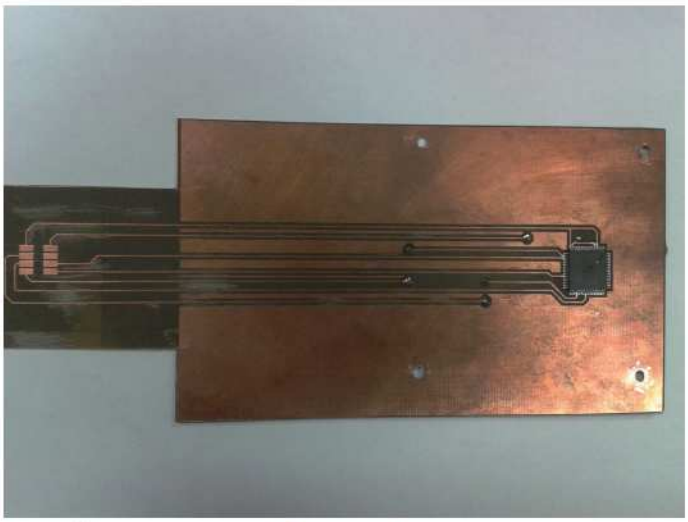
(b) Schéma d'une cellule SRAM à 6 transistors.

FIGURE 2.26: Micro-contrôleur & cellule SRAM.

tation. On peut voir que la communication avec le PC de contrôle est réalisée via le protocole carte à puce. Pour réduire les effets des différentes couches de métallisation présentes dans ce circuit (pouvant altérer l'injection de fautes et ainsi réduire la pertinence de l'analyse des différents résultats) les tirs laser sont effectués par la face arrière du circuit comme on peut le voir figure 2.27b. Celui-ci a été ouvert mécaniquement et aminci d'une centaine de μm pour obtenir des conditions d'injection optimales (*cf.* partie 1.4.2).

L'utilisation de toute la mémoire RAM du micro-contrôleur n'étant pas nécessaire pour cette étude, nous nous sommes concentrés sur quelques octets de la mémoire. La zone ainsi cartographiée a une surface de 40×40 μm^2 avec un balayage par pas de $0,5$ μm. Les cartographies sont réalisées aussi bien avec un spot laser de 1 μm que de 5 μm.

Pour chaque position de la surface cartographiée, la mémoire est d'abord placée dans un état haut, puis le tir laser est déclenché. Après quelques μs, l'état de la mémoire est lu et comparé avec l'état initialement programmé. Si une faute est détectée, la position



(a) Carte d'adaptation pour le micro-contrôleur.

(b) Vue de la face arrière ouverte du micro-contrôleur.

FIGURE 2.27: Carte de test pour le micro-contrôleur.

est ajoutée à la cartographie. Ce protocole est répété pour les deux états ("0" ou "1") de la mémoire.

2.8.3 Cartographie des zones sensibles de la mémoire RAM

La figure 2.28 correspond à la cartographie des zones sensibles de la mémoire RAM pour une puissance de $0,29\ W$, une taille de spot de $1\ \mu m$ et une durée de pulse de 50 ns correspondant à l'utilisation de la source laser 1 (*c.f.* tableau 2.1).

Sur cette figure, douze cellules mémoires SRAM sont clairement identifiables. Pour chacune de ces cellules, on remarque une zone de *Bit-set* en rouge et une zone de *Bit-reset* en bleu. Aucune faute de type *Bit-flip* n'a été obtenue. De la même manière que pour la cellule SRAM à cinq transistors étudiée dans la partie 2.5, les zones de *Bit-set* et de *Bit-reset* ne se recouvrent pas.

De plus, seulement deux zones sont sensibles, contrairement aux quatre zones attendues. Cette conclusion est possible car nous connaissons quel bit de quel octet est fauté. Cette connaissance nous a aussi permis de pouvoir cartographier l'agencement

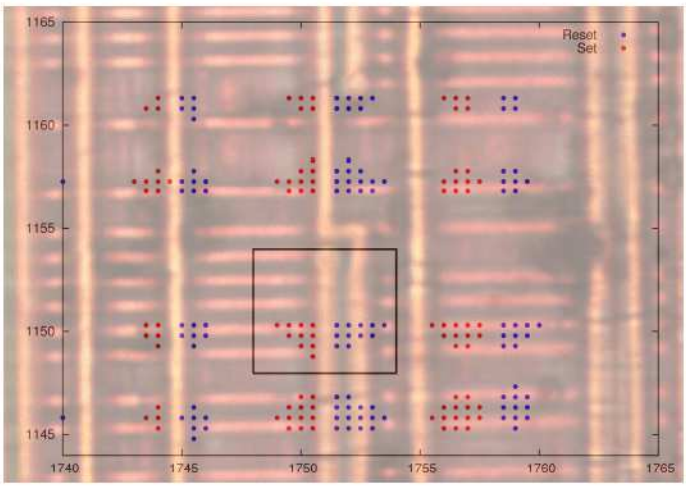


FIGURE 2.28: Cartographie des zones sensibles de la mémoire RAM avec une puissance de $0,29\ W$.

des différents bits et octets de cette mémoire. De même il nous a été facile de déduire la taille approximative d'une cellule SRAM pour ce circuit. Sur la figure 2.28, le carré représente une cellule SRAM avec une taille de $5\ \mu m \times 5\ \mu m$.

Il peut être noté que lors de ces injections de fautes, le taux de fautes mono-bit injectées (un seul bit fauté sur toutes la mémoire RAM) atteignait les 99%.

Une seconde cartographie de la mémoire est menée avec cette fois une puissance plus élevée ($0,32\ W$). Cette cartographie est présentée figure 2.29.

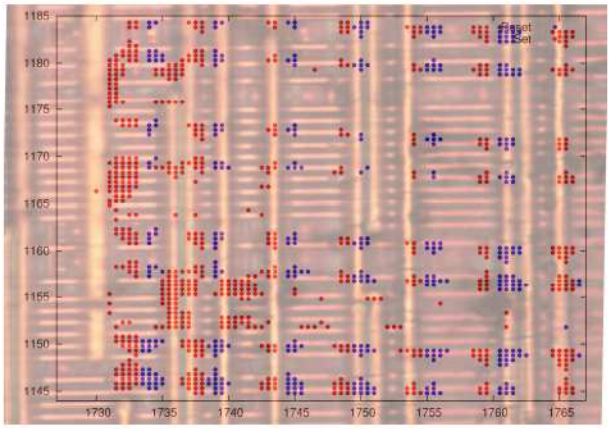


FIGURE 2.29: Cartographie des zones sensibles de la mémoire RAM avec une puissance de $0,32\ W$.

Avec cette puissance plus élevée, plus de cellule SRAM sont sensibles mais identiquement à la cartographie avec un puissance de $0,29\ W$ (figuer 2.28), les zones de *Bit-set* et de *Bit-reset* ne se recouvrent pas. Cependant, pour quelques cellules SRAM une troisième zone sensible apparaît (une seule position sensible). Cette expérience montre que l'augmentation de la puissance n'a pas d'effet sur l'apparition ou non de fautes de type *Bit-flip*.

Une troisième cartographie de la mémoire fut réalisé avec une puissance de $0,29\ W$ est une taille de spot de $5\ \mu m$. Cette fois-ci, comme on peut le voir sur la figure 2.30, les différentes cellule SRAM ne sont plus identifiables, mais ce n'était pas le but de cette dernière cartographie. En revanche, on peut observer que les zones de *Bit-set* et de *Bit-reset* ne se recouvrent pas, confirmant ainsi l'impossibilité de faire des fautes *Bit-flip*.

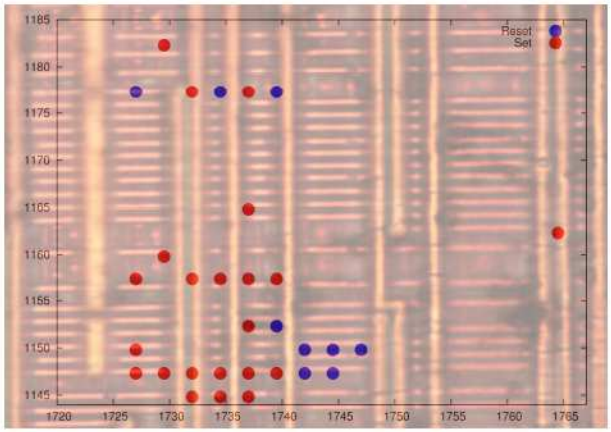


FIGURE 2.30: Cartographie des zones sensibles de la mémoire RAM avec une puissance de $0,29\ W$ et une taille de spot de $5\ \mu m$.

Avec ces trois cartographies de cette mémoire RAM et malgré l'utilisation de plusieurs paramètres différents, aucune faute *Bit-flip* n'a pu être injectée par un tir laser. Cela confirme les conclusions faites dans la partie 2.5 à partir d'injection de fautes sur une cellule mémoire ainsi que les différentes simulations d'injection de fautes laser conduites (*c.f.* 2.6).

2.8.4 Cartographie à l'aide d'une source laser picosecondes

Comme pour la cellule mémoire SRAM, des injections de fautes ont été réalisées à l'aide de la source laser 4 (*c.f.* tableaué2.1) délivrant des pulses d'une durée de 30 ps. Lors des tests sur la cellule SRAM, il a été observé que l'utilisation de pulses de cette durée permettaient de ne plus avoir de zones insensibles aux tirs laser. Ces zones insensibles ont aussi été observées sur la mémoire RAM d'un micro-contrôleur, il était donc logique de vérifier si comme pour la SRAM, ces zones identifiées comme insensibles ne l'étaient plus lorsque l'injection de fautes était réalisée avec une durée de pulse de 30 ps.

Les premiers tests ont été réalisés dans les mêmes conditions que celles utilisées dans la partie 2.8.3 : tir par la face arrière, zone cartographiée de 40 $\mu m \times 40$ μm et diamètre de spot de 1 μm. Seule l'énergie de tir est différente et égale à 2,38 nJ en sortie d'objectif.

Les résultats de cette première cartographie sont présentés figure 2.31.

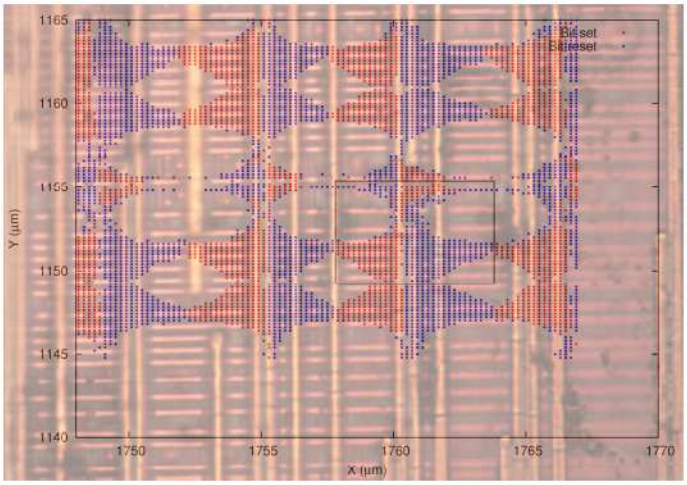


FIGURE 2.31: Cartographie des zones sensibles de la mémoire RAM avec une puissance de $2,38$ nJ et une taille de spot de 1 μm.

On peut identifier sur la figure 2.31 seize cellules SRAM. On observe cette fois, pour chaque cellule mémoire, quatre zones sensibles. Les deux zones identifiées non sensibles figure 2.28, ne le sont plus. On obtient bien le même phénomène que pour l'injection de fautes sur la cellule SRAM mis en évidence dans la partie 2.7.

Cependant, contrairement aux tests effectués avec une durée de pulse plus grande (partie 2.8.3), l'absence de fautes de type *Bit-flip* n'est pas totale. Quelques fautes de

ce type ont pu être injectées. Néanmoins, la proportion de ce type de fautes reste très faible (de l'ordre de 1%) par rapport aux fautes de type *Bit-set* ou *Bit-reset*.

De même lorsque la taille de spot est de 5 μm, les différents points mémoires ne sont plus clairement visibles (figure 2.32) mais là encore, la proportion de fautes de types *Bit-flip* reste inférieur à 1%. Cette proportion reste faible et le modèle de fautes de type *Bit-set* ou *Bit-reset* reste le plus pertinent.

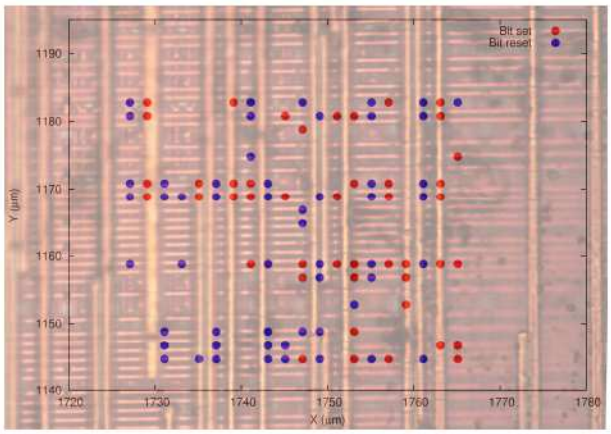


FIGURE 2.32: Cartographie des zones sensibles de la mémoire RAM avec une puissance de $1,85\ nJ$ et une taille de spot de 5 μm.

2.9 Conclusion

Les résultats d'expériences sur l'injection de fautes laser sur une cellule SRAM de configuration, semblable à celles utilisées pour mémoriser le *bitstream* de configuration des FPGA, ont été présentés. Plusieurs puissances (ou énergies), différentes tailles de spot laser ainsi que plusieurs durées de pulse laser ont été utilisées pour étudier les modèles de fautes sur ce type de circuit. Ces différentes injections de fautes ont montré que le modèle de fautes de type *Bit-flip* n'est pas pertinent pour cette cellule mémoire. Seules les fautes *Bit-set* et *Bit-reset* sont réalisables contrairement aux hypothèses de départ basées sur la capacité du spot laser à recouvrir plusieurs zones sensibles et à pouvoir obtenir des fautes indépendantes des données (*Bit-flip*).

Des simulations électriques *SPICE* ont permis de confirmer l'infaisabilité de fautes *Bit-flip* sur la cellule SRAM par un faisceau laser. Ces simulations ont aussi permis de mieux comprendre la position de transition entre la zone de *Bit-set* et *Bit-reset* et ainsi d'apporter une explication sur l'absence de fautes *Bit-flip*.

Ces résultats d'expériences et de simulations étant obtenus pour une cellule SRAM particulière, des injections de fautes laser ont été conduites sur une mémoire RAM d'un micro-contrôleur pour confirmer les résultats précédents. De même que pour une seule cellule SRAM, pour une surface contenant plusieurs cellules SRAM, le modèle de fautes *Bit-flip* n'est pas pertinent. Cependant même si quelques fautes *Bit-flip* peuvent être injectées, la proportion reste très faible par rapport aux fautes *Bit-set* ou *Bit-reset* (<1%).

Le fait de n'avoir quasiment que des fautes de types *Bit-set* ou *Bit-reset* pour de l'injection de fautes laser sur mémoire SRAM peut être critique. En effet, ce type de fautes permet de déployer assez facilement des attaques de types *Safe Error* [8] (*c.f.* partie 3.3.6) contre les systèmes cryptographiques.

Les résultats d'expériences ont de plus, mis en évidence l'absence d'une zone sensible, comme mentionné dans la partie 2.5. Les simulations *SPICE* ont permis d'expliquer cette insensibilité. Ces simulations ont permis de mettre en évidence l'importance du layout et du transistor d'accès dans l'absence de cette zone sensible. Les injections de fautes menées sur la mémoire RAM d'un micro-contrôleur ont confirmés l'absence de cette zone de sensibilité.

Cependant, lorsque la durée de pulse est de 30 ps, cette zone n'est plus insensible. Plusieurs hypothèse ont été proposées pour expliquer ce phénomène mais d'autres simulations devront être conduites avec un modèle adapté pour confirmer ces hypothèses. Cette zone insensible a été utilisée dans [57] pour améliorer la robustesse des cellules SRAM aux injections de fautes. Cependant, ces résultats expérimentaux montrent que l'utilisation de ces zones insensibles comme contre-mesures aux injections de fautes sur cellule mémoire sont moins efficaces lorsque la durée de pulse est de 30 ps.

Chapitre

3 Injection de fautes laser sur un ASIC AES

Préambule

Ce chapitre présente plusieurs attaques en fautes de la littérature sur l'algorithme de chiffrement AES *après une description détaillée des différentes transformations de cet algorithme. Les modèles de fautes induits par un tir laser sur un circuit* ASIC *dédié au chiffrement* AES *sont ensuite présentés. A l'aide des données collectées lors de l'étude des modèles de fautes sur le circuit* ASIC*, une comparaison d'efficacité de plusieurs des attaques sera présentée et réalisée en début de chapitre. La simplification d'une attaque tenant compte des modèles de fautes observés et permettant d'obtenir une meilleure efficacité sera proposée. Enfin, les modèles de fautes induits étant connus et maîtrisés, une évaluation des contre-mesures embarquées dans ce circuit* ASIC *sera menée, confirmant ainsi plusieurs hypothèses de vulnérabilité. Ces travaux ont fait l'objet de plusieurs communications dans [54, 51, 50, 55].*

Contents

3.1 Introduction

Les algorithmes de chiffrement peuvent être implémentés de manière logicielle à l'aide d'un micro-contrôleur mais aussi de manière matérielle avec des circuits spécifiques de type ASIC. Il est donc important de connaître les modèles de fautes possibles sur ces circuits ASIC dédiés au chiffrement de données.

En effet sur ce type de circuit, les contraintes lors d'un tir laser ne sont pas les mêmes que sur une cellule mémoire. Les couches de métallisation ont une influence sur les injections si le tir est réalisé par la face avant, de même si la faute est injectée dans la logique combinatoire du circuit et non directement dans un élément de mémorisation.

Pour réaliser cette étude un circuit ASIC implémentant l'algorithme AES avec une taille de clef de 128 bits a été choisi.

3.2 L'algorithme AES

Cette partie donne une description détaillée des différentes transformations utilisées par l'algorithme AES [42]. Cette description permettra de mieux comprendre les différents schémas d'attaques en fautes sur cet algorithme.

Les données manipulées sont représentées par une matrice de 4×4 octets appelée matrice d'état. La Figure 3.1 représente l'algorithme de l'AES pour une taille de clef de 128 bits. Pour la suite, tous les travaux ont été réalisés avec une taille de clef de 128 bits.

Transformation SUBBYTES

La transformation SUBBYTES est une transformation non linéaire appliquée à chacun des octets de la matrice d'état. Il s'agit d'une transformation de substitution utilisant une table de substitution appelée S-Box. La S-Box peut être exprimée mathématiquement par une combinaison de deux opérations : une transformation affine et une multiplication inverse dans $GF(2^8)$.

Le plus souvent cette transformation SUBBYTES utilise une table de substitution (*look up table*). Les valeurs des S-Box sont stockées en mémoire, pour chaque valeur d'octet

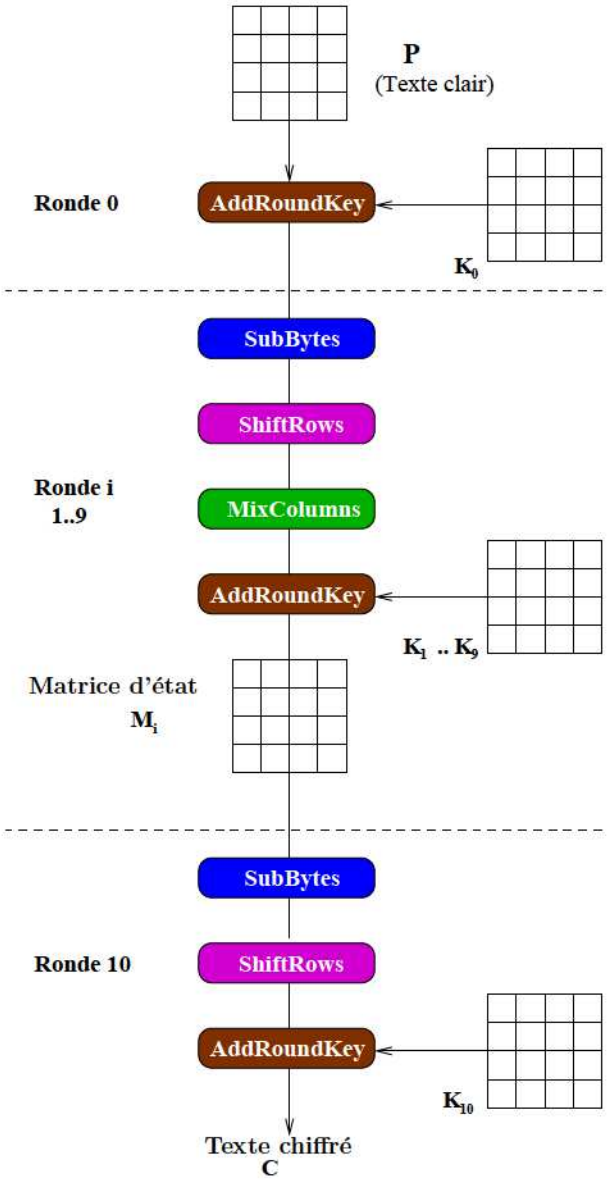


FIGURE 3.1: Représentation schématique de l'algorithme AES-128.

possible en entrée, la valeur correspondante en sortie après la transformation SUBBYTES est calculée à l'avance puis stockée dans la *look up table*. Cependant, cette transformation peut aussi être implémentée de manière à n'utiliser que de la logique combinatoire [67] et ainsi ne pas utiliser de mémoire. La transformation est alors effectuée à la volée pendant l'exécution de l'algorithme.

Transformation SHIFTROWS

La transformation SHIFTROWS consiste à faire un décalage à gauche, cyclique, des octets de chaque ligne de la matrice d'état. Aucun décalage n'est fait sur la première ligne, les octets de la deuxième ligne sont décalés d'un rang vers la gauche. Pour la troisième et la quatrème ligne de la matrice d'état, on effectue respectivement deux et trois décalages vers la gauche.

Transformation MIXCOLUMNS

La transformation MIXCOLUMNS réalise une multiplication dans $GF(2^8)$ colonne par colonne de la matrice d'état. Chaque colonne de la matrice d'état est multipliée par la matrice ci-dessous permettant d'obtenir les nouvelles valeurs de la matrice d'état après la transformation MIXCOLUMNS :

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

Transformation ADDROUNDKEY

La transformation ADDROUNDKEY est un simple "ou exclusif" bit à bit entre les octets correspondants de la matrice d'état et de la clef de ronde. La transformation KEY EXPANSION prend en entrée la clef de chiffrement et calcule pour chaque ronde de l'algorithme une clef de ronde utilisée par l'ADDROUNDKEY.

Calcul des clefs de ronde (KEY EXPANSION)

Ces calculs utilisent principalement les mêmes transformations que le chemin de données. La particularité de ces calculs est qu'ils s'effectuent colonne par colonne sur la matrice représentant les 16 octets de la clef. La première sous-clef utilisée pour la ronde initiale est la clef de chiffrement elle-même.

La figure 3.2 est un schéma représentatif du calcul des sous-clefs. Pour calculer la sous-clef courante, la sous-clef précédente est utilisée. La dernière colonne de la sous-

clef précédente subit d'abord une transformation de rotation cyclique appelée ROTWORD puis une transformation de substitution, SUBWORD utilisant la même S-Box que la transformation SUBBYTES, puis une addition bit à bit avec une constante RCON, différente pour chaque ronde de l'algorithme. Le résultat est alors additionné bit à bit avec la première colonne de la sous-clef précédente. Comme le montre la figure 3.2, les trois autres colonnes de la sous-clef sont calculées en effectuant simplement une addition bit à bit entre le résultat de la colonne précédente et la même colonne de la sous-clef précédente. Par la suite, une sous-clef pourra aussi être appelée clef de ronde.

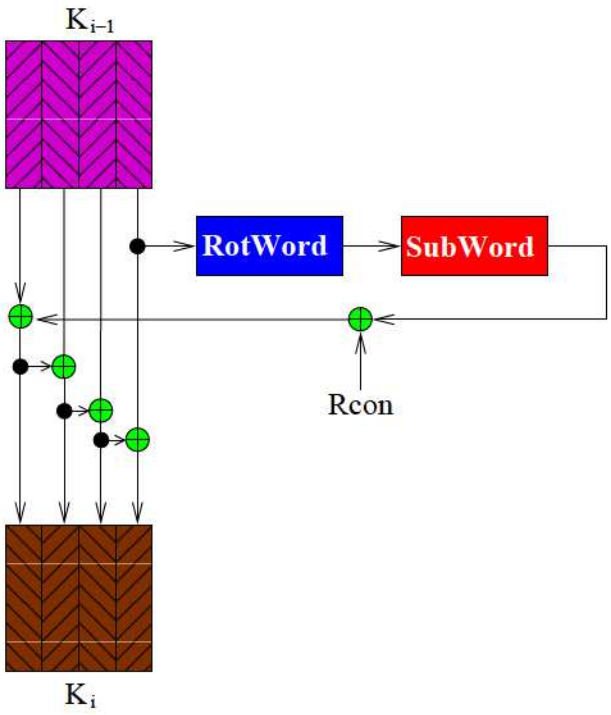


FIGURE 3.2: Représentation schématique du calcul d'une clef de ronde.

3.3 Attaques en fautes sur AES

On se concentrera ici sur les attaques en fautes pouvant utiliser comme moyen d'injection une source laser. Lorsqu'une ou plusieurs fautes sont injectées durant l'exécution d'un algorithme de chiffrement, une analyse des résultats fautés est ensuite nécessaire

pour pouvoir extraire la clef de chiffrement ou toute autre information secrète. Une des méthodes d'analyse parmi les plus utilisées est appelée *Analyse de Fautes Différentielle* (*Differential Fault Analysis : DFA*). La *DFA* a été introduite pour la première fois en 1997 [9] [7] : cette méthode consiste à comparer les chiffrés correct et fauté pour pouvoir ainsi retrouver la clef de chiffrement utilisée par l'algorithme. Depuis, beaucoup d'attaques concernant l'AES utilisant cette méthode d'analyse ont été publiées. Certaines s'intéressent aux opérations de ronde telles que le SUBBYTES [18] [31] ou le MIXCOLUMNS [44] [39], mais aussi le calcul des sous-clefs [49] [26] ou la réduction du nombre de rondes exécutées [64] [14]. Les attaques en fautes sur l'algorithme AES les plus importantes sont regroupées dans un tableau comparatif présenté dans [22]. Les attaques décrites dans la partie suivante concernent les attaques portant sur les opérations des deux dernières rondes de l'AES. Elles ne sont pas les plus efficaces en terme de textes fautés nécessaire pour retrouver la clef de chiffrement (*cf.* [22]) mais le critère de choix a été le modèle de fautes de chacune de ces attaques.

3.3.1 Notations utilisées

On note M_i la matrice d'état à la fin de la i^{eme} ronde de l'AES. P représente le texte clair en début de chiffrement et C le résultat du chiffrement de P avec la clef K. D désigne le résultat du chiffrement fauté de P avec la clef K. K_i désigne la sous-clef de la i^{eme} ronde. SB, SR, MC et ARK désignent respectivement les opérations SUBBYTES, SHIFTROWS, MIXCOLUMNS et ADDROUNDKEY. Lorsqu'une faute est injectée, sa valeur est désignée par e. Cette erreur peut être exprimée en fonction de C et D avec l'équation 3.1.

$$e = C \oplus D \tag{3.1}$$

3.3.2 Attaque sur la transformation SUBBYTES

L'attaque décrite dans [18] par Christophe Giraud utilise des fautes mono-bit devant être injectées avant la transformation SUBBYTES de la dernière ronde. Le modèle de fautes de cette attaque est donc d'obtenir des fautes mono-bit sur un ou plusieurs octets de la matrice d'état en début de dernière ronde. La figure 3.2 présente un schéma de

la dernière ronde de l'algorithme AES lorsqu'une faute est injectée sur un octet de la matrice d'état avant l'opération SUBBYTES. On remarque que la faute n'affecte que l'octet où elle a été injectée en raison de l'absence de la transformation MIXCOLUMNS durant la dernière ronde. Pour retrouver la valeur de l'octet de K_{10} correspondant à la localisation de la faute injectée, on compare alors le chiffré correct avec le chiffré fauté.

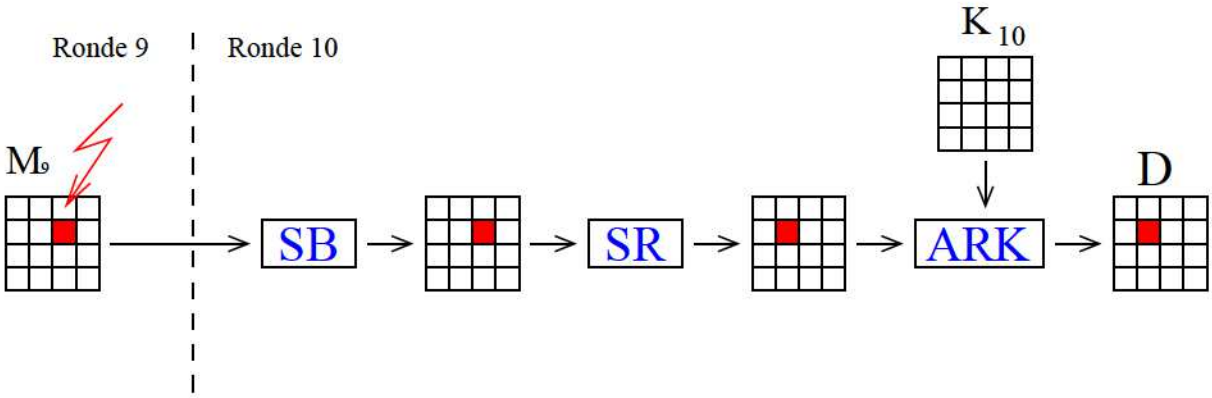


FIGURE 3.3: Schéma de l'attaque de C.Giraud [18].

L'équation 3.2 représente les différentes opérations de la dernière ronde de l'AES non fauté :

$$C = \text{SR} \circ \text{SB}(M_9) \oplus K_{10} \tag{3.2}$$

Les opérations de la dernière ronde étant effectuées octet par octet, pour plus de simplicité, la transformation SHIFTROWS peut être omise sans altérer la portée de l'analyse. On obtient une équation simplifiée pour C (equation 3.3) :

$$C = \text{SB}(M_9) \oplus K_{10} \tag{3.3}$$

Pour le chiffré fauté, on obtient (équation 3.4) :

$$D = \text{SB}(M_9 \oplus e) \oplus K_{10} \tag{3.4}$$

En appliquant un OU-EXCLUSIF entre les équations 3.3 et 3.4, on obtient successivement les équations 3.5 et 3.6 :

$$\Delta = C \oplus D \tag{3.5}$$

$$\Delta = \text{SB}(M_9) \oplus \text{SB}(M_9 \oplus e) \tag{3.6}$$

L'attaque reposant sur le fait que les fautes injectées sont mono-bit, il existe 2040 (8×255) couples (M_9, e) possibles. Pour chaque couple possible, Δ est calculé puis comparé avec la valeur de Δ trouvée pour le couple de chiffrés correct/fauté. On obtient alors un ensemble de valeurs possibles pour (M_9, e) contenant la valeur correcte pour M_9. Si cet ensemble contient plusieurs valeurs, on les teste alors de la même manière avec un second couple de chiffrés correct/fauté. L'opération est recommencée avec d'autres chiffrés correct/fauté jusqu'à ce que l'ensemble des valeurs possibles pour (M_9, e) ne contienne plus qu'une seule possibilité. Ayant trouvé la valeur de M_9, l'équation 3.7 permet de calculer la valeur de K_{10}.

$$K_{10} = \text{SB}(M_9) \oplus C \tag{3.7}$$

Pour retrouver la clef K, il suffit alors de faire l'opération de calcul inverse des clefs de ronde (INV KEY EXPANSION) une fois que tous les octets de K_{10} ont été retrouvés.

Selon [18], cette attaque est efficace à 97% avec trois couples de chiffrés correct-fauté (C, D) pour retrouver un octet de la sous clef K_{10}.

3.3.3 Attaque de Roche et al.

Cette attaque vise le module de calcul des clefs de ronde de l'algorithme. Une faute doit être injectée sur la clef de l'avant-dernière ronde (ronde 9) de sorte que K_9 et K_{10} soient affectées par cette faute, on a alors :

$$\tilde{K_9} = K_9 \oplus E_9 \tag{3.8}$$

$$\tilde{K_{10}} = K_{10} \oplus E_{10} \tag{3.9}$$

où $\tilde{K_9}$ et $\tilde{K_{10}}$ sont les deux dernières clefs de ronde fautées. E_9 et E_{10} sont les valeurs des fautes des deux dernières clefs de ronde. L'attaque présentée en 2011 [49] se déroule alors en plusieurs étapes. On commence par chiffrer N messages différents sans injection de fautes. On recommence ensuite avec les mêmes messages mais cette fois-ci une faute est injectée sur le calcul de la sous-clef de la ronde 9. On obtient N paires de chiffrés correct/fauté (C,D). L'analyse de ces paires chiffrés correct/fauté étant réalisée octet par octet, l'opération SHIFTROWS peut être omise. Les équations 3.10 et 3.11 représentent

les calculs de la dernière ronde pour C et D.

$$C = SB(M_9) \oplus K_{10} \tag{3.10}$$

$$D = SB(M_9 \oplus E_9) \oplus K_{10} \oplus E_{10} \tag{3.11}$$

À partir de 3.10 et 3.11 on peut exprimer D avec l'équation 3.12 :

$$D = SB(SB^{-1}(C \oplus K_{10}) \oplus E_9) \oplus K_{10} \oplus E_{10} \tag{3.12}$$

Pour chaque hypothèse possible du triplé (E_9, E_{10}, K_{10}) notée (e_9, e_{10}, k_{10}), un compteur T est associé à l'octet correspondant. Pour chaque paire de chiffrés correct/fauté, on incrémente de 1 la valeur du compteur T associé à chaque triplé si l'équation 3.13 est vérifiée.

$$SB(SB^{-1}(C \oplus k) \oplus e_9) \oplus k \oplus e_{10} = D \tag{3.13}$$

Si les fautes injectées sur la sous-clef de la ronde 9 sont reproductibles, les valeurs de E_9 et E_{10} sont constantes pour tous les couples de chiffrés correct/fauté. Il n'y a alors qu'un seul triplé (e_9, e_{10}, k_{10}) dont la valeur du compteur est N. Il correspond aux valeurs correctes de (E_9, E_{10}, K_{10}). Dans le cas où les fautes injectées sont constantes, il faut alors trois paires de chiffrés correct/fauté pour obtenir un taux de réussite de 90%. En revanche lorsque les fautes injectées ne sont pas constantes, le nombre de paires chiffrés correct/fauté pour mener l'attaque à son terme va alors augmenter en fonction de la variabilité des fautes injectées. Comme le montre la figure 3.4 extraite de [49], lorsque le taux de reproductibilité de la faute injectée décroit, le nombre de paires de chiffrés correct/fauté nécessaire pour réussir l'attaque va alors augmenter exponentiellement.

Pour cette attaque, le modèle de fautes utilisé est différent de l'attaque présentée précédemment (partie 3.3.2). Le modèle de fautes est donc d'obtenir une faute sur un ou plusieurs octets lors du calcul de la sous-clef de la ronde 9. De plus, les fautes injectées lors de plusieurs calculs de chiffrement doivent être d'une valeur la plus constante possible, l'idéal étant d'avoir des valeurs de fautes constantes dans 100% des cas. Cette dernière contrainte sur le modèle de fautes implique d'avoir une certaines maîtrise sur les bits fautés.

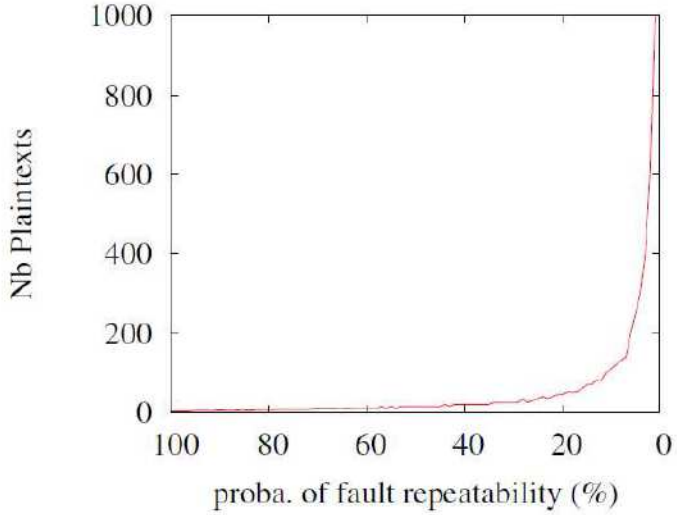


FIGURE 3.4: Nombre de paires chiffrés correct/fauté nécessaires pour un taux de réussite de 90% en fonction du taux de répétabilité des fautes injectées [49].

3.3.4 Attaque de Lashermes et al.

Cette attaque, présentée en 2012 par Lashermes et al. [31], s'intéresse à la non uniformité de la distribution des valeurs de fautes injectées en début de dernière ronde, avant la transformation SUBBYTES. La seule contrainte sur le modèle de fautes associé à cette attaque est d'obtenir une distribution des valeurs de fautes non-uniforme. En 2011, Rivain et al. [47] ont présenté une attaque similaire sur l'algorithme de chiffrement symétrique DES.

La dernière ronde de l'AES ne comportant pas la transformation MIXCOLUMNS, l'analyse des chiffrés correct/fauté peut être réalisée octet par octet. L'analyse présentée par la suite sera sur un octet.

Si une erreur est injectée au début de la dernière ronde, on obtient alors les deux equations 3.14 et 3.15 pour les chiffrés correct et fauté :

$$C = SR \circ SB(M_9) \oplus K_{10} \tag{3.14}$$

$$D = SR \circ SB(M_9 \oplus e) \oplus K_{10} \tag{3.15}$$

L'analyse étant réalisée octet par octet, pour plus de simplicité, la transformation

SHIFTROWS peut être omise sans altérer la pertinence de l'analyse. Les équations 3.14 et 3.15 deviennent alors :

$$C = SB(M_9) \oplus K_{10} \tag{3.16}$$

$$D = SB(M_9 \oplus e) \oplus K_{10} \tag{3.17}$$

La première étape de cette attaque consiste à exprimer l'erreur injectée à partir des deux équations précédentes. On obtient donc pour l'erreur injectée l'équation 3.18 suivante :

$$e = SB^{-1}(C \oplus K_{10}) \oplus SB^{-1}(D \oplus K_{10}) \tag{3.18}$$

Pour chaque couple de chiffrés correct/fauté d'indice j et pour toutes les hypothèses de K_{10} possibles, notées k, on peut calculer la valeur de l'erreur injectée correspondante $e_{(j,k)}$. Cela permet de construire un tableau appelé table d'erreurs présenté tableau 3.1. Chaque colonne de ce tableau regroupe, pour une valeur de clef possible, la valeur d'erreur correspondante pour chaque couple de chiffrés correct/fauté.

TABLE 3.1: Table d'erreur

	Hypothèse de K_{10} notée k				
Réalisation j	'0x00'	'0x01	'0x02'	$\cdots$	'0xFF'
0	$e_{0,0}$	$e_{0,1}$	$e_{0,2}$	$\cdots$	$e_{0,255}$
1	$e_{1,0}$	$e_{1,1}$	$e_{1,2}$	$\cdots$	$e_{1,255}$
2	$e_{2,0}$	$e_{2,1}$	$e_{2,2}$	$\cdots$	$e_{2,255}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Une seule colonne du tableau 3.1 correspond alors à la valeur correcte de l'octet visé de K_{10}. Les erreurs calculées dans cette colonne correspondent aux erreurs effectivement injectées.

Pour distinguer l'hypothèse correcte de K_{10} des autres hypothèses fausses, l'attaque se base sur une distribution des valeurs de fautes injectées non uniforme. Pour les mauvaises hypothèses de clef, les erreurs calculées vont avoir une distribution aléatoire, alors que l'hypothèse correcte aura une distribution biaisée due à la non uniformité de la distribution des valeurs de fautes réellement injectées.

Cette attaque utilise comme distingueur l'entropie de Shannon, l'équation 3.19 donne son expression :

$$H(p_s) = -\sum_{e=0}^{255} p_s(e) * \log_2 p_s(e) \tag{3.19}$$

où $p_s(e)$ est la probabilité d'occurrence de la valeur de l'erreur e (entre 1 et 255) pour une hypothèse de clef k par rapport au nombre de réalisation j. Pour les mauvaises hypothèses de clef, la distribution des valeurs de l'erreur étant aléatoire, l'entropie de Shannon va tendre vers 8. En revanche, pour la bonne hypothèse de clef, l'entropie va être inférieure à 8, ce qui permet de la différencier des autres hypothèses.

Le nombre minimum de couples chiffrés correct/fauté nécessaire pour pouvoir distinguer la bonne hypothèse de clef des mauvaises hypothèses est discuté dans [31] et illustré avec la figure 3.5

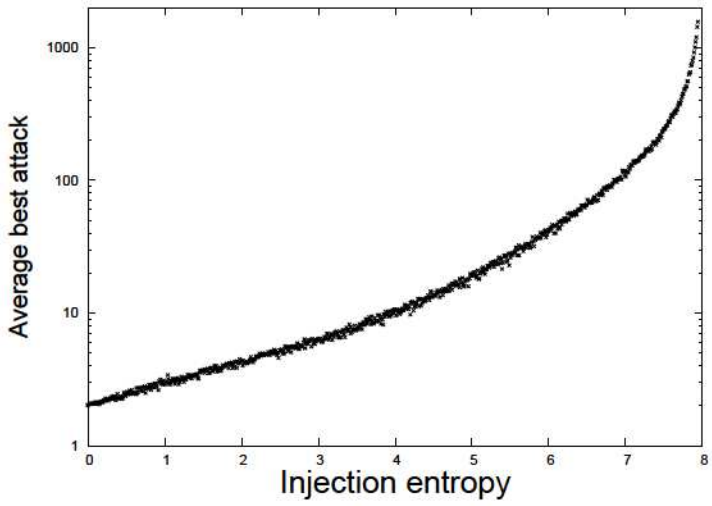


FIGURE 3.5: Nombre moyen de textes nécessaires à l'attaque de Lashermes et al. en fonction de l'entropie des fautes injectées [31].

On remarque bien avec la figure 3.5 que plus les valeurs de fautes injectées sont identiques (entropie plus basse) plus le nombre de couples chiffrés correct/fauté nécessaires diminue (avec un minimum de 2) et facilite la distinction entre la bonne hypothèse de clef et les autres hypothèses erronées.

3.3.5 Attaque sur la transformation MIXCOLUMNS

Cette attaque présentée en 2003 par Piret et al. [44], impose des contraintes moins fortes sur le type de fautes que l'attaque sur le SUBBYTES présentée dans la partie 3.3.2. En effet, la faute doit ici être mono-octet et injectée avant la transformation MIXCOLUMNS de la ronde 9. Lorsqu'une faute est injectée en respectant ces contraintes, on obtient un schéma de propagation de la faute à travers l'algorithme de chiffrement similaires à la Figure 3.6.

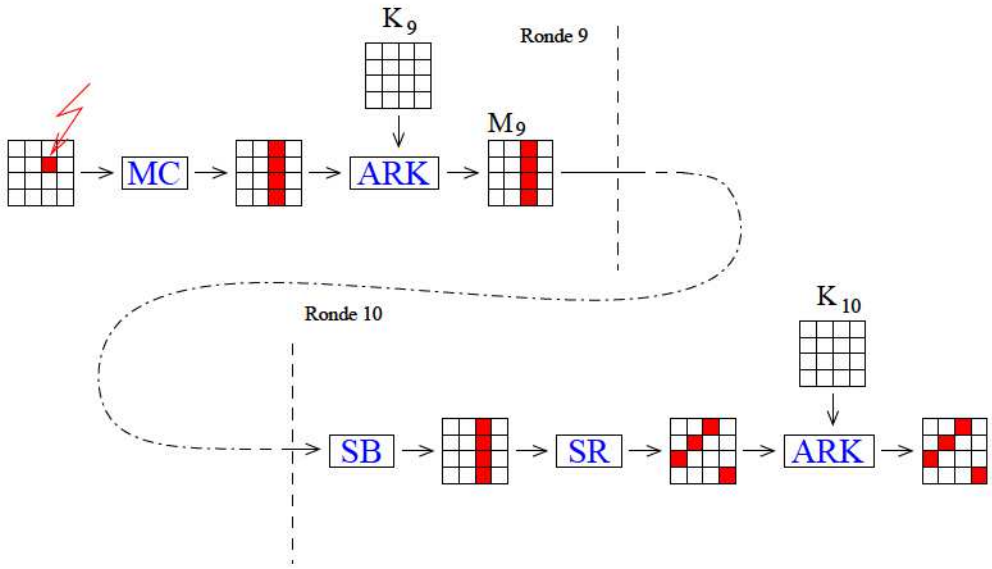


FIGURE 3.6: Schéma de l'attaque de G. Piret et al. [44].

Cette attaque permet de retrouver quatre octets de la clef de ronde K_{10} avec seulement un octet fauté. Cela est possible grâce à l'opération MIXCOLUMNS qui propage la faute injectée sur tous les octets de la colonne de la matrice d'état, comme on peut l'observer sur la figure 3.6.

La première partie de l'attaque consiste à créer une liste de toutes les valeurs possibles des quatre octets fautés à la fin de le 9^{ieme} ronde. La colonne de la matrice d'état fautée après le MIXCOLUMNS est connue par analyse de la position des quatre octets fautés de D. Néanmoins, la position dans cette colonne de l'octet fauté avant le MIXCOLUMNS est inconnue, on a donc 1020 (4×255) possibilités dans cette liste. On appelle cette liste G. Une seconde liste L contient toutes les valeurs possibles des quatre octets de

K_{10} correspondant aux quatre octets fautés. Cependant, pour limiter la complexité des calculs, pour le début de l'analyse, on considère dans la liste L seulement les valeurs possibles de deux octets de K_{10}. La liste L contient donc 2^{16} valeurs possibles de deux octets de K_{10} (noté k). Avec C et D, on calcule pour chaque hypothèse de L la valeur E de la faute en début de ronde 10 avec l'équation 3.20 :

$$E = SB^{-1}(C \oplus k) \oplus SB^{-1}(D \oplus k) \tag{3.20}$$

Si la valeur de E n'est pas contenue dans G, on retire la valeur k de L et on passe à la valeur suivante. Une fois que toutes les valeurs de L ont été testées , la taille de l'ensemble L est significativement réduite. On étend ensuite la liste avec le troisième octet de K_{10} et on recommence. On fait ensuite de même avec le quatrième octet. On obtient à la fin les valeurs correctes pour les quatre octets de K_{10} correspondant aux quatre octets fautés.

Pour avoir une efficacité de 99%, deux couples de chiffrés correct/fauté sont nécessaires. Avec cette attaque, on doit donc injecter une faute sur un octet de chaque colonne de la matrice d'état pour pouvoir retrouver les 16 octets de la clef de chiffrement.

Dans ce même article, les auteurs proposent d'injecter une faute une ronde plus tôt dans le déroulement de l'algorithme. En effet, si une faute est injectée avant la transformation MIXCOLUMNS de la $8^{ième}$ ronde, elle va alors se propager sur toute la colonne de la matrice d'état. On aura une faute sur un octet de chaque colonne de la matrice d'état avant le dernier MIXCOLUMNS. A la fin du chiffrement, les 16 octets seront fautés. En appliquant l'analyse présentée précédemment, on retrouve alors les 16 octets de la clef de chiffrement.

En injectant une faute avant le MIXCOLUMNS de la $8^{ième}$ ronde, 2 couples de chiffrés correct/fauté sont nécessaires pour atteindre un taux de réussite de 77%.

3.3.6 Attaque de type *Safe Error*

Ce type d'attaque a été présenté initialement en 2000 par Yen et al. [68] portant sur un algorithme de chiffrement à clef publique. Cette attaque fut portée en 2003 par Blömer et al. [8] sur l'algorithme de chiffrement à clef secrète AES et restreint à une taille de clef de 128 bits.

Cette attaque nécessite une injection de fautes immédiatement après l'exécution de la ronde initiale de l'algorithme. De plus la faute doit être de type *Bit-set* ou *Bit-reset*. Lors de cette ronde initiale, seul la transformation ADDROUNDKEY est réalisée avec la clef de chiffrement :

$$M_0 = P \oplus K \tag{3.21}$$

Si le texte à chiffrer est choisi comme ayant tous ses bits valant 0, l'équation de la ronde initiale devient :

$$M_0 = 0 \oplus K \tag{3.22}$$

$$M_0 = K \tag{3.23}$$

L'attaquant va alors injecter une faute sur un bit donné suivant le modèle de fautes *Bit-reset* par exemple. La valeur du bit va être forcée à 0. Si le résultat du chiffrement est faux, cela signifie que le bit visé de la clef était égal à 1 avant l'injection de la faute. En revanche, si le résultat du chiffrement est correct, on en déduit que la valeur du bit visé de la clef est égal à 0. L'opération doit être répétée sur chaque bit de la clef pour retrouver sa valeur entière.

Cette attaque peut aussi être menée avec des fautes injectées suivant le modèle *Bit-set*. La valeur du bit visé sera alors forcée à 1.

Ce type d'attaque permet facilement d'exploiter les modèles de fautes observées dans les parties 2.5 et 2.8 lors de tirs laser sur une cellule mémoire SRAM et plus particulièrement sur une mémoire RAM d'un micro-contrôleur. Si la clef et/ou les sous-clefs sont stockées dans cette mémoire RAM, le modèle de fautes étant maîtrisé et correspondant à des fautes de type *Bit-set* ou *Bit-reset*, un bit de la clef ou d'une sous-clef peut être forcé à 1 ou 0, permettant par la suite de déduire la valeur initiale du bit fauté conformément à la description de cette attaque.

3.4 L'ASIC AES

Le circuit utilisé pour les différents tests d'injection a été présenté en 2011 dans [3]. Le circuit a été réalisé avec la technologie *CMOS* 130 nm de STMicroelectronics avec six couches de métallisation. Il fonctionne à une fréquence de 25 MHz et a une tension

d'alimentation de 3, 3 V. La communication avec l'extérieur se fait via une interface APB 32 bits alors que les chemins de données sont de 128 bits. La taille totale de la puce est de 1336 μm $\times$ 1411 μm. Une photo du circuit en face avant est donnée figure 3.7.

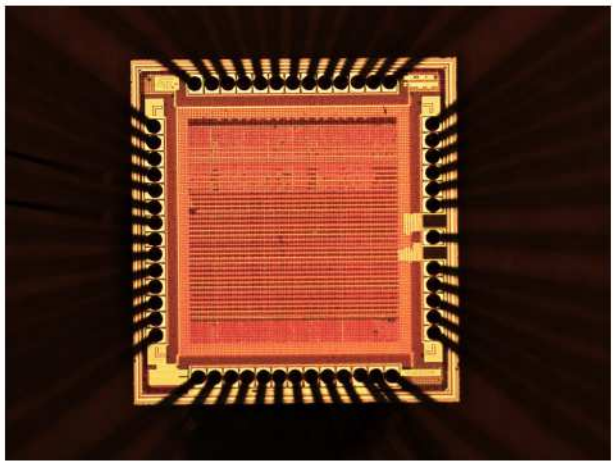


FIGURE 3.7: Photo en face avant de l'ASIC AES.

L'implémentation de l'algorithme AES est entièrement matérielle. De plus, la transformation SUBBYTES n'utilise pas de *Look-up table* pour les S-Box, elle est entièrement réalisée en logique combinatoire comme décrit dans [67]. La transformation SUBBYTES sur un octet, peut être vue comme une substitution dans $GF(2^8)$, qui est un sous corps de $GF(2^4)$. On peut donc réaliser cette substitution en logique combinatoire comme le montre le schéma de la figure 3.8.

De plus, le circuit comporte plusieurs contre-mesures contre les attaques en fautes [3, 23]. Le chemin de données est tout d'abord dupliqué : en plus du chemin de données normal, un deuxième chemin de données est présent utilisant les données complémentaires de celles fournies par l'utilisateur. Cette redondance des chemins de données permet de comparer les données manipulées et de détecter d'éventuelles erreurs. Si une erreur est détectée sur un octet, celle-ci est ensuite propagée sur plusieurs octets. Cette opération de détection et propagation est insérée pendant l'opération SUBBYTES. Enfin, pendant l'opération SHIFTROWS, plusieurs bits des deux chemins de données sont croisés. Ces contre-mesures permettent de contrer deux types d'attaques. Tout d'abord les attaques en fautes en détectant l'injection de fautes. Lorsqu'une faute est détectée, l'erreur est diffusée sur la matrice d'état afin d'obtenir une confusion des résultats fautés

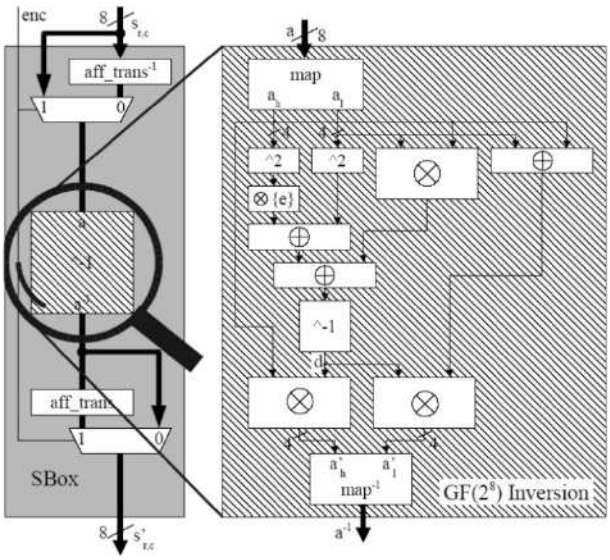


FIGURE 3.8: Schéma de l'operation SUBBYTES. [67]

et ainsi rendre difficile voir impossible l'analyse de ces résultats. Ces contre-mesures permettent aussi de protéger le circuit contre les attaques par canaux auxiliaires. En effet, le second chemin de données complémentées permet d'avoir une consommation d'énergie (ou une émanation électromagnétique) globale neutre, que ce soit en poids ou distance de Hamming, évitant ainsi d'avoir des différences selon les données manipulées. Un schéma d'ensemble de l'architecture de l'ASIC est donné à la figure 3.9.

Sur la gauche, on peut observer l'interface APB établissant une communication 32 bits avec l'extérieur et mettant en forme les données en entrée et sortie pour la partie du circuit réalisant les chiffrements AES. La partie réalisant le chiffrement peut être séparée en trois parties. La première partie concerne la machine d'état qui contrôle le déroulement du chiffrement. La deuxième réalise les calculs des sous-clefs à la volée. En effet, les sous-clefs ne sont pas calculées à l'avance. En réalité, il y a deux blocs réalisant les calculs de sous-clefs, un bloc pour les sous-clefs du chemin de données normales plus un bloc pour les sous-clefs du chemin de données complémentées. La dernière partie comprend les quatre opérations utilisées lors d'une ronde de l'AES. Les deux chemins de données se rencontrent lors de l'opération SUBBYTES pour la détection et la propagation d'éventuelles erreurs ainsi que lors de l'opération SHIFTROWS croisée.

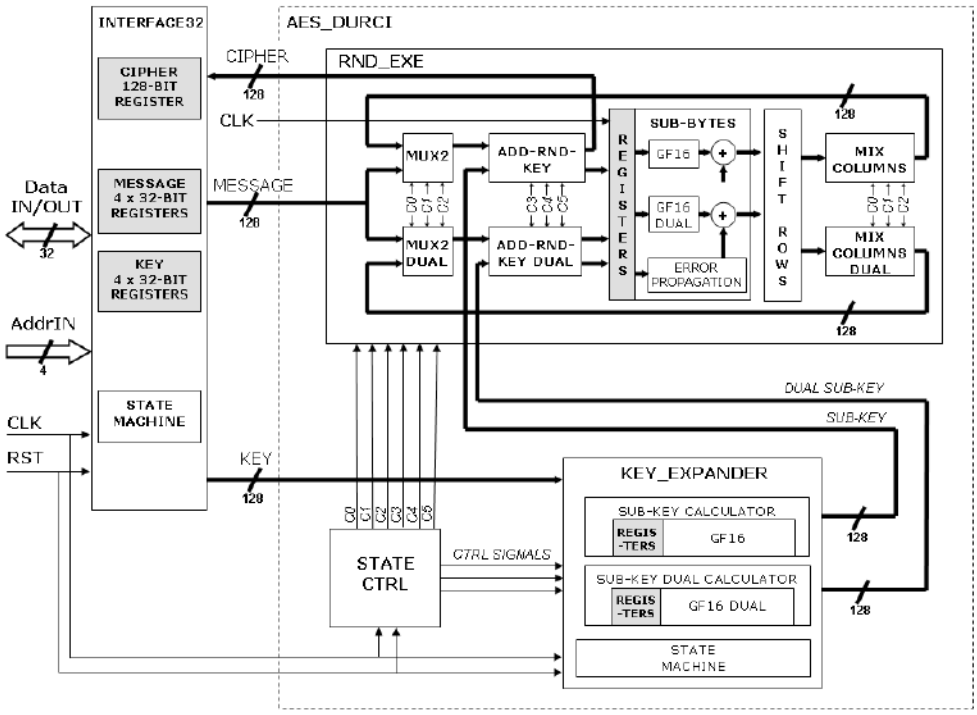


FIGURE 3.9: Schéma de l'ASIC AES.

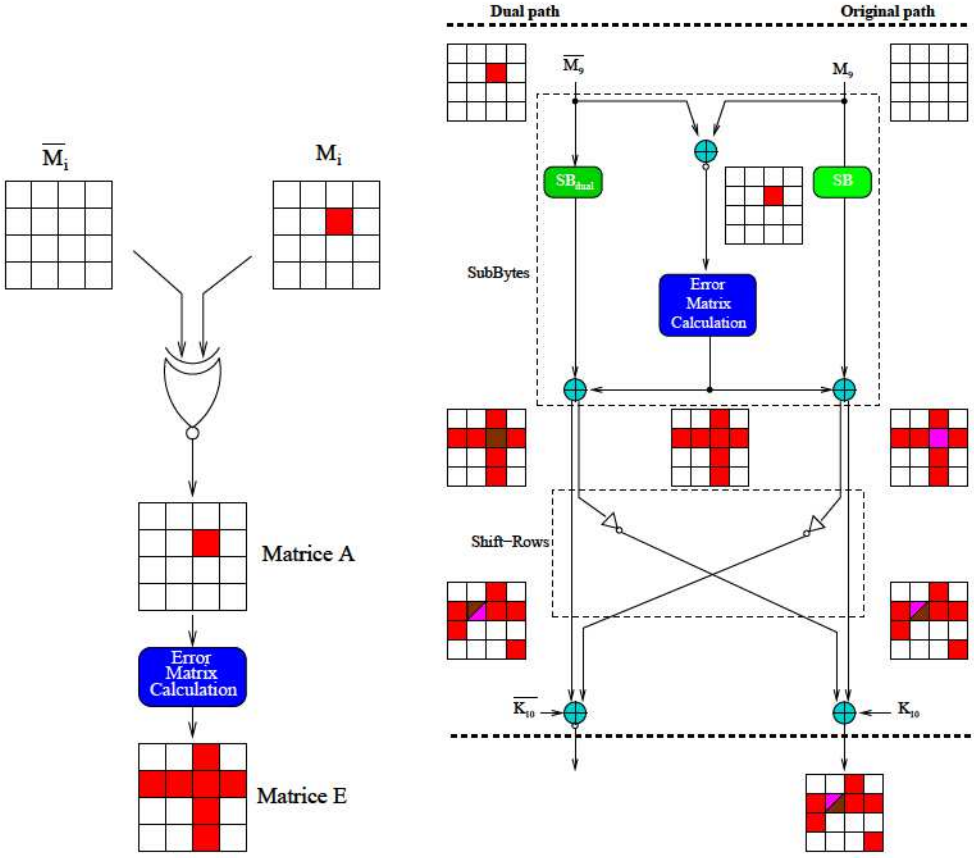
3.4.1 Contre-mesures

Comme mentionné précédemment, le circuit embarque plusieurs contre-mesures contre les attaques en fautes. La détection et propagation d'erreurs consiste à réaliser une opération de NON-OU-EXCLUSIF (XNOR), octet par octet, entre les matrices d'états des deux chemins de données avant le début de chaque ronde. Les deux chemins de données étant complémentaires, le XNOR permet de vérifier la complémentarité des chemins et donc de détecter une erreur. Si une erreur est détectée, elle est étendue sur la ligne et la colonne relatives à sa position comme illustré avec la figure 3.10a. Chaque élément a_{ij} de la matrice A aura comme valeur $a_{ij} = 0$ sauf l'élément correspondant à la position de l'octet fauté où on aura $a_{ij} = e$. Le calcul de chaque élément e_{ij} de la matrice E se fait alors à l'aide des équations 3.24 et 3.25. Une fois la matrice E calculée, celle ci est

propagée sur chacun des deux chemins de données.

$$r_i = \sum_{j=0}^{3} a_{ij} \; et \; c_j = \sum_{i=0}^{3} a_{ij} \tag{3.24}$$

$$e_{ij} = r_i \oplus c_j \tag{3.25}$$



(a) Détection et propagation d'erreur.

(b) Propagation d'une faute lors de la dernière ronde de l'AES.

FIGURE 3.10: Illustration des contre-mesures matérielles du circuit AES.

En plus du système de détection et de propagation, lors de la transformation SHIFTROWS, pour chaque octet, la moitié des bits sont croisés avec ceux du chemin de données complémentées. La figure 3.11 illustre un exemple, de croisement de bit entre les deux chemin de données. Chacun des 16 octets a un croisement des bits entre les deux chemins de données différent. En présence d'une faute, cela a pour effet de cacher à

l'attaquant une partie de l'information sur la faute injectée. En effet, à la fin du chiffrement, celui-ci n'a accès qu'au chiffré du chemin de données normal. La figure 3.10b permet d'avoir une idée du fonctionnement du mécanisme de détection et propagation des erreurs lorsqu'une faute est injectée sur l'un des deux chemins de données avant la transformation SUBBYTES de la dernière ronde de l'AES.

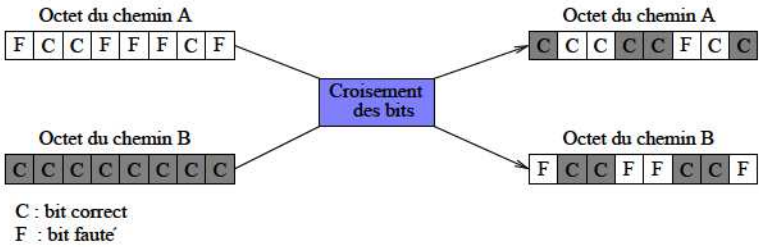


FIGURE 3.11: Exemple de croisement des bits d'un octet entre les deux chemins de données.

Il faut ajouter que dans le but d'empêcher un attaquant de pouvoir identifier visuellement les différents blocs constituant l'AES, la logique du circuit a été lors de sa conception éparpillée sur toutes la surface du circuit. On a donc un circuit où aucun motif n'est reconnaissable visuellement. On verra par la suite que ce choix ne sera pas sans conséquence lors d'injection de fautes.

3.4.2 Carte et Banc de test

Les différents tests d'injection de fautes sur le circuit AES ont été réalisés sur le banc d'injection laser de la plate-forme collaborative Micropacks [37]. Le banc laser, présenté avec la figure 3.14b et différent du banc laser utilisé dans le chapitre 2, utilise une source laser *YAG* (*Yttrium Aluminium Garnet*) offrant le choix entre trois longueurs d'ondes : 355 nm (ultraviolet), 532 nm (vert) et 1064 nm (infrarouge). Pour chaque longueur d'onde, la taille du spot peut être réglée via un obturateur rectangulaire. Celui-ci permet d'avoir des tailles de spot allant de 0 (faisceau laser totalement obstrué) à un faisceau laser carré de 2500 μm de coté. Le faisceau ainsi retaillé passe à travers un objectif permettant d'obtenir un spot laser focalisé utilisable pour l'injection de fautes. De même l'énergie du laser peut être réglée entre $0,15$ μJ et $0,5$ mJ. Une table motorisée

permet de fixer la carte de test et ainsi de réaliser des campagnes d'injections en plusieurs positions. Cette table offre une résolution sur les axes X et Y de 0,1 μm.

Un PC de contrôle permet de paramétrer le laser (puissance de tir, taille de spot, instant de tir) mais aussi d'envoyer au circuit les données nécessaires pour effectuer les chiffrements et récupérer les données chiffrées. Le PC contrôle aussi une carte de synchronisation permettant d'ajouter un délai entre le signal de déclenchement provenant du circuit et le tir laser. Ce délai permet de respecter les temps de chauffe imposés par le laser. Au final, la précision de l'instant de tir atteint $10ns$. Un oscilloscope peut aussi être ajouté pour "monitorer" la consommation du circuit.

La figure 3.12 montre les différentes interactions entre les équipements du banc d'injection lors d'une campagne de test. Le PC de contrôle configure dans un premier temps le laser, la carte de synchronisation ainsi que la position de la table XY. Puis le PC démarre la communication avec le circuit test. Ceci a pour effet de déclencher le signal de synchronisation fourni par le circuit. Un premier signal est alors envoyé au laser depuis la carte de synchronisation pour démarrer le temps de chauffe du laser puis le signal de tir est envoyé au laser. Une fois le tir laser effectué, le PC récupère la réponse du circuit test aux commandes envoyées au démarrage ainsi que les courbes d'acquisition de l'oscilloscope. Le PC met alors à jour les différents paramètres pour continuer la campagne d'injection.

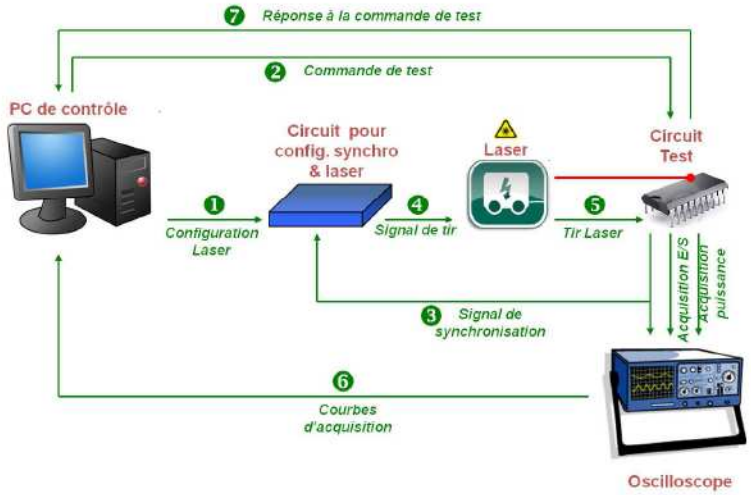


FIGURE 3.12: Schéma du banc de test laser.

L'observation de la consommation du circuit testé pendant l'injection de fautes a une double utilité. En effet, cela permet d'observer clairement l'effet d'un tir laser sur la consommation mais aussi de vérifier la bonne synchronisation entre le tir laser et l'exécution de l'algorithme de chiffrement. Un exemple de consommation électrique du circuit test à deux instants d'injection différents est donné figure 3.13. On observe clairement les différentes rondes de l'AES ainsi qu'un pic de consommation lors du tir laser. On peut vérifier aisément que le tir a bien eu lieu pendant la 7^{ieme} ronde de l'AES pour la courbe de couleur verte et durant la 8^{ieme} ronde de l'AES pour la courbe de couleur rouge.

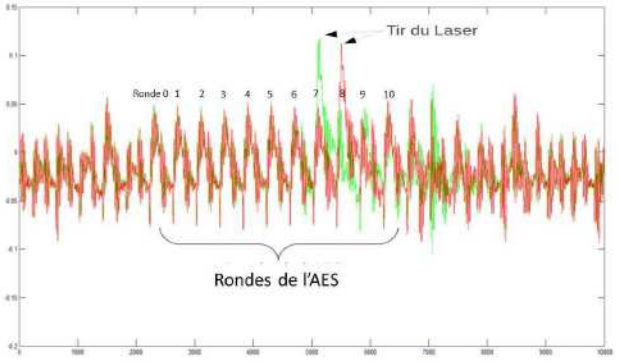
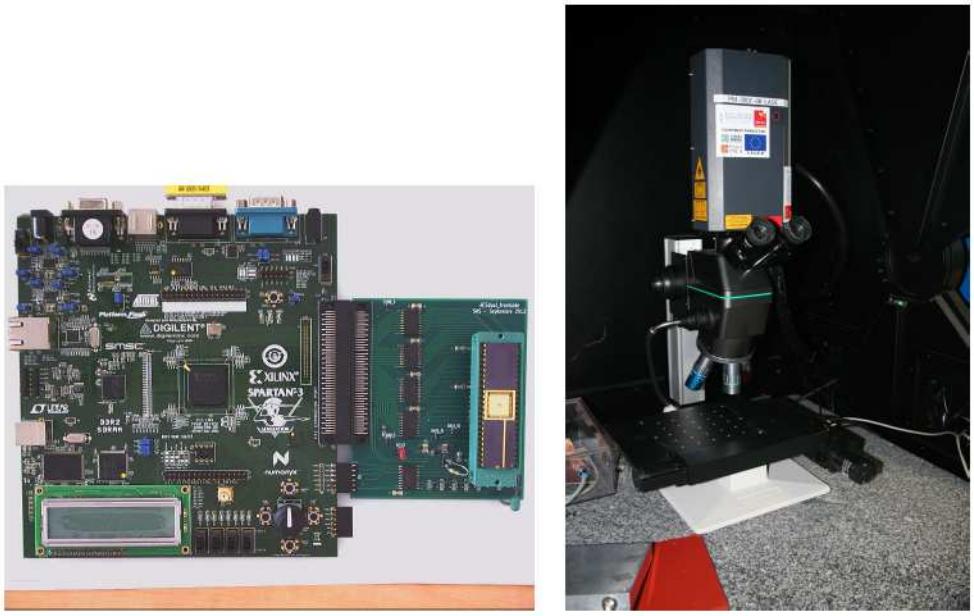


FIGURE 3.13: Consommation électrique du circuit lors d'un tir laser pendant la 7^{ieme} et la 8^{ieme} ronde de l'AES-128.

Le circuit ne peut pas être utilisé seul pour les expérimentations, une carte d'adaptation est nécessaire. Comme on peut le voir sur la figure 3.14a, cette carte en deux parties est composée d'une carte FPGA Xilinx Spartan 3 connectée avec une carte fille recevant le circuit. La carte fille, en plus de réaliser la connexion du circuit avec le FPGA, permet de mettre en forme les signaux de contrôle et de données provenant du FPGA pour communiquer avec le circuit AES. Le FPGA permet de réaliser plusieurs fonctions. Tout d'abord l'interface entre le circuit test et le PC de contrôle via une communication série RS-232 mais aussi la génération du signal de trigg envoyé à la carte de synchronisation. Ce signal de trigg est généré à partir d'une commande envoyée par le PC de contrôle au FPGA.



(a) Carte d'adaptation pour le circuit AES. (b) Banc laser.

FIGURE 3.14: Carte support pour l'ASIC & banc laser.

3.5 Étude du modèle de fautes en face avant

Comme expliqué dans la partie 1.4.2, l'injection peut se faire soit par la face avant, soit par la face arrière. Chacune de ces méthodes possède ses avantages et inconvénients. Pour rappel, l'injection par la face arrière nécessite une préparation du circuit cible qui peut parfois être compliquée à mettre en place mais procure un accès aux zones sensibles sans les couches de métallisation faisant écran. L'injection de fautes par la face avant, nécessite moins de préparation et est souvent plus rapide à mettre en place. En revanche, les différentes couches de métal recouvrent les zones sensibles. Avec les technologies CMOS avancées, on peut se poser la question de la pertinence de cette méthode d'injection par la face avant.

De plus, la taille de spot laser minimum étant limitée à 1 μm et la taille des transistors continuant à décroître, l'hypothèse faite que le faisceau laser n'atteint qu'une zone sensible devient discutable, de même que le modèle de fautes associé : *Bit-set/Bit-reset*. Le modèle de faute *Bit-flip* pourrait sembler plus pertinent.

Le but de cette étude est multiple ; il peut être scindé en deux parties. La première partie s'attache à vérifier quel modèle de fautes est pertinent et réalisable sur un circuit avec un tir laser par la face avant avec une taille de spot large. Cette configuration permet de se placer dans le cas d'une injection laser bas coût : accès seulement à la face avant et un équipement d'injection laser bas coût ne possédant pas des objectifs évolués autorisant une petite taille de spot et donc, a priori, rendant difficile de satisfaire aux modèles de fautes mono-bit et mono-octet. La deuxième partie de l'étude s'intéresse naturellement à l'exploitation des fautes injectées et présente une comparaison de plusieurs schémas d'attaques en fautes classiques au vue des particularités des fautes injectées.

3.5.1 Conditions expérimentales

Pour cette étude, la taille du faisceau laser utilisé était de 125 × 125 μm^2, soit l'ouverture la plus large que le banc laser puisse offrir avec un objectif x*20*. Pour chaque tir, l'énergie était positionnée à 750 nJ, on obtient alors, après traversée du chemin optique, une densité d'énergie de 17 $pJ/\mu m^2$. La surface du circuit est divisée en 36 zones comme le montre la figure 3.15. Chaque zone correspond au positionnement du faisceau laser durant l'injection de fautes. La synchronisation du tir laser avec l'exécution de l'algorithme AES par le circuit était paramétrée de façon à ce qu'une faute soit injectée avant l'opération SUBBYTES de la dernière ronde.

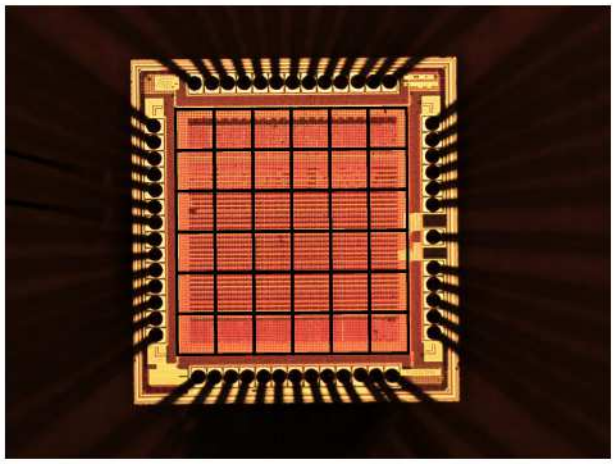


FIGURE 3.15: Surface de l'ASIC AES divisé en 36 zones.

3.5.2 Analyse des modèles de fautes

Les 36 zones délimitées de la surface du circuit test ont été dans un premier temps soumises au faisceau laser. Pour chaque position, 10000 chiffrements ont été réalisés à partir de textes clairs aléatoires. À chaque fois qu'un chiffré était fauté, connaissant le chiffré correct ainsi que la clef de chiffrement, il était facile de retrouver la faute injectée et le type de faute à partir des équations 3.26 et 3.27 :

$$e = SB^{-1}(C \oplus K_{10}) \oplus SB^{-1}(D \oplus K_{10}) \tag{3.26}$$

$$M_9 = SB^{-1}(C \oplus K_{10}) \tag{3.27}$$

À partir de l'équation 3.26, on peut remonter à la valeur de la faute injectée et ainsi déterminer le nombre de bits impactés par l'injection de fautes. L'équation 3.27 permet quant à elle de remonter à la valeur de M_9 et ainsi d'avoir la valeur initiale du ou des bits fautés. On peut alors déterminer le type de fautes injectées et une éventuelle dépendance aux données. On obtient alors pour chaque octet le taux d'injection de fautes, c'est-à-dire le nombre de chiffrés fautés sur les 10000 chiffrements, le taux d'injection de fautes mono-bit par rapport au taux d'injection, ainsi que le taux de répétabilité de la faute. Ces résultats sont présentés dans le tableau 3.2. À chaque octet correspond une position spatiale du spot laser sur la surface du circuit. On peut remarquer ici que l'éparpillement des différents blocs logiques sur la surface du circuit n'a pas eu d'effet significatif sur l'injection de fautes. Les résultats présentés pour chacun des octets correspondent à la localisation ayant conduit aux taux d'injection de fautes le plus élevé. On a donc seize positions différentes du spot laser sur la surface du circuit permettant d'obtenir les résultats du tableau 3.2.

Le taux de répétabilité d'une faute représente le taux d'occurrence d'une même faute sur cet octet. Par exemple pour l'octet 0, sur 10000 chiffrements, 480 ont été fautés (soit un taux d'injection de fautes de 4,8%). Sur ces 480 chiffrés fautés, 79% (379) l'étaient par des fautes mono-bit et 74% (355) l'étaient par la même valeur de faute.

Après une première analyse des résultats, on remarque que le taux d'injection de fautes est relativement faible pour tous les octets (sauf l'octet 3). Le taux de fautes mono-bit ainsi que le taux de répétabilité sont eux en revanche particulièrement élevés. Pour avoir plus d'informations sur le type des fautes injectées, des tirs laser supplémentaires

TABLE 3.2: Résultats expérimentaux sur ASIC AES.

Octet #	Taux d'injection de fautes	Taux de fautes mono-bit	Répétabilité
0	4,8%	79%	74%
1	3,2%	100%	99%
2	3,1%	98%	92%
3	67,8%	49%	48%
4	9,4%	99,7%	90%
5	2,1%	79%	58%
6	0,5%	100%	99%
7	4,6%	65%	64%
8	23%	64%	42%
9	7,2%	91%	80%
10	4,3%	99%	98%
11	15,5%	97%	97%
12	12,2%	98%	96%
13	3,1%	87%	55%
14	0,2%	100%	100%
15	7%	99,2%	99%

ont été réalisés à la position spatiale correspondant à l'injection de fautes sur l'octet 5. Cette fois-ci, 1000 textes clairs aléatoires ont été utilisés. Les résultats obtenus sont résumés dans le tableau 3.3.

TABLE 3.3: Injection de faute Laser sur l'octet 5 (1000 textes aléatoires).

Taux d'injection de fautes	Taux d'apparition de la faute '0x80'	Taux d'apparition d'autres fautes
7.1%	94%	6%

On retrouve bien avec ces injections que le taux de fautes est faible (7,1%) mais que le taux d'occurrence de fautes mono-bit avec une valeur sur un octet de $0x80$ est très élevé (94%). En s'intéressant plus précisément à la valeur du bit fauté avant l'injection, on remarque que les seules fois où le bit est fauté, sa valeur initiale était "0". Aucune faute n'a été injectée lorsque sa valeur initiale était "1". On a donc une faute de type

Bit-set. Dans ce cas, si on considère seulement les fautes de type *Bit-set* pour l'injection de fautes, le taux d'injection atteint alors 14.2% (les textes sont aléatoires).

Lorsqu'une campagne d'injection est effectuée sur l'octet 5 avec un texte clair choisi, permettant l'injection de fautes de type *Bit-set* (valeur initial du bit fauté à "0"), les différents taux d'injections montrent un léger écart avec les 14% attendus comme le montre le tableau 3.4 mais confirme bien le modèle de fautes considéré.

TABLE 3.4: Injection de fautes laser sur l'octet 5 (texte choisi unique).

Taux d'injection de fautes	Taux d'apparition de la faute '0x80'	Taux d'apparition d'autres fautes
16.8%	97%	3%

En revanche, le taux d'injection obtenu de 16% reste très bas compte tenu du fait que l'injection de fautes par laser est généralement considérée comme déterministe. Un explication possible est la dépendance de l'injection de fautes laser à l'instant d'injection (*c.f.* partie 1.2.4). En effet, le banc laser introduit un jitter de 10 ns sur l'instant de tir effectif, sachant qu'une période d'horloge est de 40 ns. Ce jitter important par rapport à la période d'horloge laisse donc penser que certains tirs laser provoquent un SET mais que celui-ci ne provoque pas de faute.

Parmi le grand nombre de données récoltées lors de la première campagne d'injections (chiffrement de 10000 textes aléatoires), une analyse plus poussée a été effectuée sur l'octet 3. Plus précisément, une étude bit à bit des fautes injectées sur cet octet est rapportée dans le tableau 3.5.

La colonne *Valeur de la faute* reporte la valeur de e calculée à partir de l'équation 3.26. On remarque alors facilement que les quatre bits de poids fort ne sont jamais fautés mais aussi que le bit 1 (b_1) a un taux d'injection deux fois plus élevé que le bit 2 (b_2) (respectivement 66% et 34.3%). Á l'aide de l'équation 3.27, pour chaque injection de faute, la valeur de chacun des deux bits (b_1 et b_2) avant l'injection a pu être retrouvée. Ces informations ont permis de construire pour chacun de ces deux bits, le tableau 3.6 regroupant les probabilités d'avoir une faute provoquant le passage du bit de "0" vers "1" et de "1" vers "0".

TABLE 3.5: Fautes injectées sur l'octet 3.

Taux d'injection	Répétabilité
67.8%	48%
Valeur de la faute $b_7...b_4$ $b_3b_2b_1b_0$	Nombre d'injection
0000 0110	3285
0000 0010	3228
0000 1110	93
0000 1000	70
0000 0100	51
0000 0001	40
0000 1001	13
0000 0011	4

TABLE 3.6: Probabilité sur les fautes affectant les bit 1 et 2

	$P(0 \rightarrow 1)$	$P(1 \rightarrow 0)$
Bit 1	0,487	0,513
Bit 2	0,986	0,014

On remarque avec ce tableau que pour le bit 1, les fautes sont de type *Bit-flip* (probabilités quasi-équivalentes d'un passage de "0" vers "1" ou de "1" vers "0"). En revanche le bit 2 n'est principalement affecté que par des fautes de type *Bit-set* (probabilité d'un passage de "1" vers "0" négligeable). Cette analyse minutieuse nous montre ici que les deux modèles de fautes *Bit-set* (ou *Bit-reset*) et *Bit-flip* sont possibles avec un tir laser par la face avant et un spot large. Une explication possible sur la présence de fautes de type *Bit-set* (ou *Bit-reset*), contrairement aux hypothèses faites en début d'étude, est l'influence des différentes couches de métal qui agissent comme une fenêtre découpant le faisceau laser. Lorsque le faisceau laser atteint les différentes pistes métalliques d'une couche, celui-ci est réfléchi par le métal. Le faisceau laser va donc traverser la couche de pistes métalliques aux seuls endroits où il n'y a pas de pistes. Le circuit de test comportant six couches de métal, le faisceau laser ayant au départ une grande surface est fortement diminué après avoir traversé toutes les couches. Ce faisceau découpé par

les différentes couches ne va alors atteindre dans certains cas que des zones sensibles relatives aux fautes de type *Bit-set* ou *Bit-reset*. Dans d'autres cas, pour un même bit, les zones sensibles correspondant aux fautes de types *Bit-set* et *Bit-reset* ne sont pas recouvertes par les couches de métallisation (*c.f* partie 1.4.2). Elles sont donc accessibles, pour une même position, par le faisceau laser et permettraient donc l'injection de fautes de type *Bit-flip*.

3.5.3 DFA sur la dernière ronde de l'AES

Dans cette section, les données collectées dans la partie 3.5.2, sont utilisées pour mener différents types d'attaques sur la dernière ronde de l'AES. Le but est d'étudier l'efficacité de telles attaques avec nos données expérimentales (taux d'injection de fautes faible mais taux de répétabilité et de fautes mono-bit élevé).

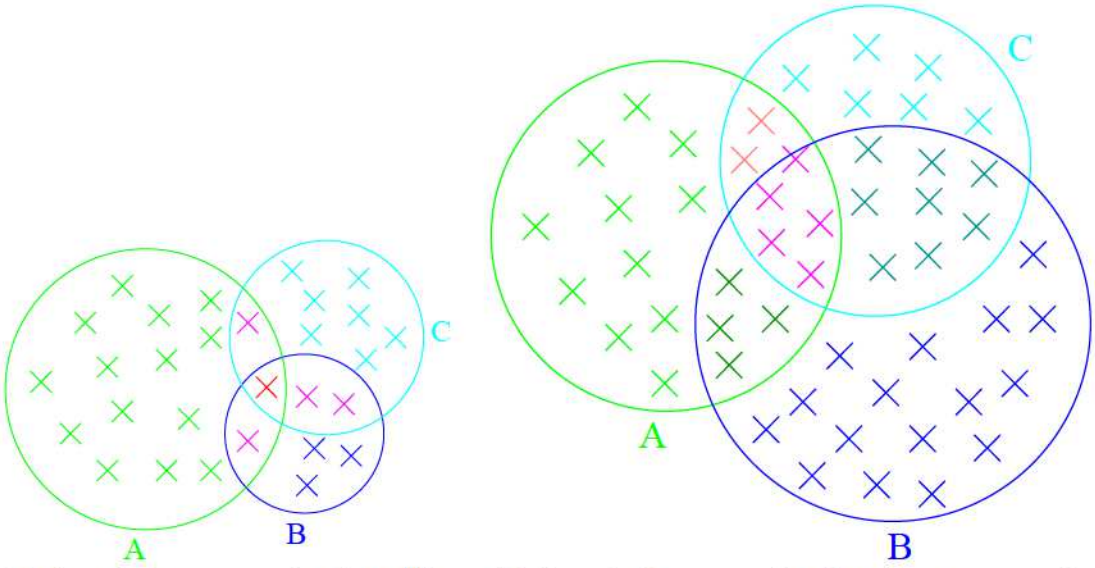
Application de l'attaque sur l'opération SUBBYTES

La première attaque menée à l'aide de nos données expérimentales pour retrouver la clef secrète, fut l'attaque de Giraud, présentée dans la partie 3.3.2, nécessitant l'injection de fautes avant l'opération SUBBYTES de la dernière ronde. Comme précisé dans la description de l'attaque, les contraintes sur les fautes sont de pouvoir injecter une faute mono-bit entre la fin de la transformation MIXCOLUMNS de la ronde 9 et le début de la transformation SUBBYTES de la ronde 10.

Les différents résultats d'injection de fautes présentés dans le tableau 3.2 ont donc été utilisés. On remarque que seuls trois octets respectent parfaitement les contraintes sur les fautes injectées de cette attaque. Les octets 1, 6 et 14 ont été fautés par des fautes mono-bit mais le taux d'injection de fautes pour ces octets est extrêmement bas et pas assez significatif pour affirmer que trois paires de chiffrés correct/fauté sont donc nécessaires pour retrouver chacun des octets 1, 6 et 14 de la clef de ronde K_{10} avec un taux de succès de 97%.

En revanche, 13 octets présentent des taux d'injection de fautes suffisamment significatif pour être exploité. Ces octets présentent des taux d'injection de fautes mono-bit proches de 80%. Le nombre de paires de chiffrés correct/fauté alors nécessaires pour retrouver les octets correspondants de K_{10} augmente. En effet, comme l'illustre la fig-

ure 3.16a, lorsque la contrainte sur le type de fautes est respectée, la bonne valeur de l'octet de sous-clef recherchée correspond à l'intersection des ensembles des valeurs de sous-clefs possibles pour trois couples de chiffrés correct/fauté. En revanche, lorsque l'on a comme dans notre cas, un taux de fautes mono-bit proche des 80% et si on utilise seulement trois couples de chiffrés correct/fauté alors on obtient un sous ensemble de valeurs possibles ne permettant pas de conclure sur la valeur correct de la sous-clef. Il faut donc utiliser plus de couples chiffrés correct/fauté pour pouvoir obtenir une seule hypothèse de sous-clef correcte.



(a) Taux de fautes mono-bit de 100% et trois couple de chiffrés correct/fauté.

(b) Taux de faute mono-bit de 80% et trois couples de chiffrés correct/fauté

FIGURE 3.16: Illustration des ensembles de valeurs possibles de l'octet de sous-clef recherché lors de l'attaque de Giraud.

On peut calculer le nombre minimum de couples chiffrés correct/fauté pour obtenir un taux de réussite de l'attaque d'au moins 90% lorsque le taux d'injection de fautes mono-bit est de 80%.

Les injections de faute étant indépendantes les unes des autres, on peut modéliser le nombre de fautes mono-bit obtenues par une variable aléatoire X qui suit une loi binomiale de paramètres n, le nombre de fautes effectuées, et $\frac{4}{5}$, la probabilité d'obtenir une faute mono-bit.

Le but est d'obtenir au moins 3 couples de chiffrés correct/fauté avec une faute mono-bit pour pouvoir réussir l'attaque. Supposons que l'on ait au moins 5 couples de chiffrés correct/fauté.

$$P(X \geq 3) = \sum_{i=3}^{5} P(X = i) \tag{3.28}$$

$$= \binom{5}{3} \left(\frac{4}{5}\right)^3 \left(\frac{1}{5}\right)^2 + \binom{5}{4} \left(\frac{4}{5}\right)^4 \left(\frac{1}{5}\right) + \left(\frac{4}{5}\right)^5 = 0.94208 \tag{3.29}$$

Si l'on a 5 couples de chiffrés correct/fauté, on a plus de 94 % de chances d'obtenir au moins 3 couples de chiffrés correct/fauté avec une faute mono-bit (*cf.* équation 3.29).

Afin de calculer la probabilité de réussir l'attaque, considérons les événements suivants :

- Soit A l'événement réussir l'attaque.
- Soit B l'événement obtenir exactement 3 couples de chiffrés correct/fauté avec une faute mono-bit.
- Soit C l'événement obtenir au moins 3 couples de chiffrés correct/fauté avec une faute mono-bit.

La probabilité calculée dans [18], est la probabilité de réussir l'attaque sachant que l'on a exactement 3 fautes mono-bit soit $P_B(A)$. La probabilité de réussir l'attaque augmentant avec le nombre de fautes mono-bit obtenues, on a $P_B(A) \leq P_C(A)$ d'où :

$$P(A) \geq P(A \cap C) = P_C(A) \cdot P(C) \geq P_B(A) \times P(C) \approx 0,91 \tag{3.30}$$

Il faut au minimum 5 couples de chiffrés correct/fauté pour obtenir un taux de réussite de cette attaque de 90% lorsque le taux d'injection de faute mono-bit est de 80%

En revanche, pour les octets 3, 7 et 8, avec des taux d'injection de fautes mono-bit respectivement de 50%, 65% et 64%, le nombre de paires chiffrés correct/fauté nécessaires devient important. Pour ces octets présentant des taux d'injection de ce type, cette attaque ne semble pas être la plus efficace.

Application de l'attaque de Roche et al.

Cette attaque, présentée dans la partie 3.3.3, est basée sur la capacité de l'attaquant à injecter des fautes constantes sur la clef de ronde K_9. Les fautes étant injectées lors du calcul des clefs de ronde, K_{10} est donc aussi affectée par les fautes injectées.

Si les fautes injectées ont un taux de répétabilité de 100%, seulement trois paires de chiffrés correct/fauté sont alors nécessaires pour retrouver l'octet attaqué de K_{10} avec un taux de succès de 90%. Cette attaque peut aussi être appliquée avec des injections de fautes ayant des taux de répétabilité inférieurs à 100% (notre cas), le nombre de paires de chiffrés augmente alors au fur et à mesure que le taux de répétabilité décroit (*cf.* figure 3.4).

Nos données expérimentales utilisées ici concernent des fautes injectées entre la fin de la transformation MIXCOLUMNS de la ronde 9 et le début de la transformation SUBBYTES de la ronde 10. K_{10} n'est donc affectée par aucune faute. Cependant, le principe de l'attaque est quand même applicable avec nos données. En effet, l'équation 3.13 devant être vérifiée pour considérer une hypothèse k de K_{10} comme potentiellement correcte, il suffit alors de considérer que l'erreur e_{10} est nulle dans l'équation 3.31.

$$SB(SB^{-1}(C \oplus k) \oplus e_9) \oplus k \oplus e_{10} = D \tag{3.31}$$

Avec les données du tableau 3.2, 9 octets (1, 2, 4, 6, 10, 11, 12,14, 15) ont un taux de répétabilité supérieur ou égal à 90% et nécessitent donc au plus 6 paires de chiffrés correct/fauté pour retrouver l'octet correspondant de la clef de ronde K_{10}. Pour les trois octets (0, 7, 9) avec un taux de répétabilité supérieur à 60%, les paires de chiffrés correct/fauté nécessaires vont alors être d'un maximum de 9. En revanche, pour les 4 octets 3, 5, 8 et 13, ayant un taux de répétabilité proche des 50% (voire plus faible), la réussite de l'attaque nécessite dans ce cas au minimum 15 paires de chiffrés correct/fauté.

De la même manière que pour l'attaque sur la transformation SUBBYTES, cette attaque permet de retrouver la clef secrète malgré la nécessité d'obtenir plus de couples chiffrés correct/fauté pour réussir l'attaque.

Cependant, cette attaque impose moins de contraintes sur le type de fautes injectées et peut permettre de réussir là où l'attaque de Giraud échouerait en raison des ces contraintes sur le type de fautes injectées (fautes mono-bit).

Simplification de l'attaque de Lashermes et al.

Les attaques précédentes permettent de retrouver relativement aisément une partie de la clef secrète. Néanmoins pour certains octets, particulièrement ceux avec un taux d'injection de fautes mono-bit ou un taux de répétabilité faibles, ces deux attaques ne sont pas efficaces. L'attaque de Lashermes et al., décrite dans la partie 3.3.4, exploite la capacité à injecter des fautes ayant une distribution des valeurs d'erreurs non uniformes. Cette non uniformité de la distribution de valeurs de fautes injectées permet de discriminer la valeur correcte de la sous-clef de l'octet fauté. Lorsque le taux de répétabilité est faible (entre 20 et 30%), cette attaque reste efficace. Elle permet de trouver la bonne hypothèse sur l'octet de la sous-clef avec seulement une dizaine de couples de chiffrés correct/fauté. Il semble donc que cette attaque soit plus efficace que les deux autres attaques utilisées plus tôt, au vu du taux de répétabilité approchant les 50%. Cette attaque peut être particulièrement efficace pour les fautes ayant des taux de répétabilité inférieurs à 50%. En utilisant les données expérimentales concernant l'injection de fautes sur l'octet 3, la table d'erreur a été construite et reportée dans le tableau 3.7 :

TABLE 3.7: Table d'erreur de l'octet # 3.

	Hypothèses de K10 noté k					
Realisation i	'0x00'	'0x01'	···	**'0xCD'**	···	'0xFF'
0	'0x63'	'0x61'	···	**'0x02'**	···	'0x15'
1	'0xB2'	'0x0A'	···	**'0x06'**	···	'0x59'
2	'0x0C'	'0xBF'	···	**'0x02'**	···	'0x1E'
···	···	···	···	···	···	···
158	'0x51'	'0xFF'	···	**'0x06'**	···	'0x1A'
···	···	···	···	···	···	···
3,578	'0xF2'	'0x49'	···	**'0x08'**	···	'0x82'
···	···	···	···	···	···	···
10,000	'0x09'	'0x3B'	···	**'0x0E'**	···	'0x33'

Normalement, comme décrit dans la partie 3.3.4, pour chaque hypothèse de clef, l'entropie des erreurs est calculée et permet de distinguer la valeur de clef correcte. Dans notre cas, il est facile d'affirmer que la valeur correcte de l'octet correspondant de K_{10} est "0xCD". Pour les mauvaises hypothèses de clef, les valeurs d'erreurs calculées semblent aléatoires. En revanche, pour l'hypothèse de clef correspondant à la valeur

"0xCD", les erreurs sont restreintes aux quatre bits de poids faibles. Cela correspond bien aux erreurs de l'octet 3 décrites dans le tableau 3.5. Il n'est donc pas nécessaire de calculer l'entropie des erreurs pour chaque hypothèse de clef car la discrimination visuelle est suffisante et efficace. Dès lors que les fautes injectées suivent un motif reconnaissable, cela permet d'alléger, en termes de calculs, l'analyse des résultats.

Un peu plus de 3 paires de chiffrés correct/fauté seulement sont nécessaires pour retrouver la bonne hypothèse de clef. En effet, l'entropie des fautes injectées sur l'octet 3 est égale à 1,3. En se reportant à la figure 3.5, on trouve qu'en moyenne 3,5 paires de chiffrés correct/fauté sont nécessaires pour réussir l'attaque. Pour les deux attaques précédentes, avec un taux de répétabilité ou de fautes mono-bit inférieur à 50%, le nombre moyen de paires de chiffrés correct/fauté nécessaires pour retrouver la valeur de l'octet de la sous-clef se situait autour de 15.

Cette simplification d'attaque permet d'être beaucoup plus efficace pour des taux de répétitions de fautes faibles et d'exploiter de possibles motifs de répétitions des fautes injectées. En effet, dans le cas des données exploitées dans cette partie, seule la moitié des bits de l'octet 3 était affectée par le spot laser (principalement du fait de la dispersion de la logique de l'ASIC AES) ce qui permet d'obtenir des fautes sur cet octet n'affectant que la moitié des bits et d'avoir un motif reconnaissable facilement. Les différentes couches de métallisation peuvent aussi permettre d'obtenir des motifs de fautes n'affectant qu'une seule partie des bits d'un octet. De même, la dépendance aux données de certains bits fautés peut aussi constituer un motif de discrimination pour cette attaque. Les erreurs calculées dans la table d'erreur correspondant à la bonne hypothèse de clef mettront en évidence toujours les mêmes bits fautés.

3.5.4 Conclusion

Cette partie a permis de montrer que malgré l'utilisation d'une taille de spot large (125×125 μm^2) pour effectuer des tirs par la face avant, des fautes de type *Bit-set* et *Bit-reset* ont pu être injectées. Ces deux types de fautes étaient inattendus du fait de la différence entre la taille du spot laser utilisé et la taille des transistors. De plus, ces types de fautes (*Bit-set* et *Bit-reset*) demandent de pouvoir illuminer seulement une zone sensible du transistor, ce qui semblait à première vue plus approprié à une taille

de spot de 1 μm. Cependant, des fautes de type *Bit-flip* ont aussi pu être injectées et correspondraient à l'illumination de plusieurs zones sensibles par le spot laser.

Une explication possible de la présence de fautes de type *Bit-flip* mais aussi de type *Bit-set*/*Bit-reset*, malgré l'utilisation d'un spot laser large, peut être trouvée dans l'action des différentes couches de métal agissant comme une fenêtre de découpe du faisceau laser et permettant de n'atteindre que certaines zones sensibles des transistors [5]. L'injection de fautes dans la logique combinatoire pourrait être une autre explication à la présence des ces types de fautes. Cependant cette possibilité est à prendre avec précaution du fait de la présence d'un jitter de 10 ns sur l'instant de tir du LASER pour un fonctionnement du circuit à une fréquence de 25 MHz. Cette étude a aussi permis de corroborer la dépendance aux données de l'injection laser.

Dans un second temps, les données expérimentales recueillies ont permis de confronter l'efficacité des attaques développées par Giraud [18] ainsi que Roche et al. [49]. On a montré qu'avec un taux d'injection de fautes mono-bit et/ou un taux de répétabilité faibles, ces deux attaques ne sont pas les plus efficaces. Une simplification de l'attaque de Lashermes et al. [31] a permis d'avoir une attaque plus efficace avec ces valeurs de taux d'injection.

3.6 Caractérisation de l'ASIC AES protégé

Comme présenté dans la partie 3.4, le circuit implémentant l'algorithme AES embarque plusieurs contre-mesures ayant pour but de protéger le circuit contre les attaques en fautes. Une partie du travail de cette thèse fut de caractériser et d'évaluer l'efficacité de ces contre-mesures vis-à-vis de l'injection de fautes par laser.

3.6.1 Étude théorique des contre-mesures

Avant d'injecter des fautes à l'aide du banc laser, une première analyse des contre-mesures et de leur comportement supposé vis à vis de l'injection de fautes permet d'établir une stratégie d'attaque et de déceler d'éventuelles failles.

Pour notre circuit, cinq analyses peuvent être faites pour décrire son comportement supposé vis-à-vis de l'injection de fautes.

Injection de fautes dans un des deux chemins de données

La première analyse est la plus logique, lorsqu'une faute est injectée directement sur le chemin de données normale. Si une faute est injectée sur le chemin de données normales on se retrouve dans le cas typique de détection de fautes par la contre-mesure. Comme décrit dans la partie 3.4.1 avec les figures 3.10b et 3.10a, la faute va être propagée sur les deux chemins de données. De plus une partie de l'information est perdue à cause de la deuxième contre-mesure. L'opération SHIFTROWS croisée entre les deux chemins de données va donc intervertir 4 bits de chaque octet de la matrice d'état des deux chemins de données. Par rapport à la valeur réelle de l'erreur après la transformation SUBBYTES, 4 bits sont faux et peuvent conduire à considérer comme correctes des hypothèses de sous-clef fausses.

Cependant, l'attaque de Giraud [18] est toujours possible. Cette attaque, comme décrit dans la partie 3.3.2, nécessite l'injection de fautes mono-bit entre la fin de la transformation MIXCOLUMNS de la ronde 9 et le début de la transformation SUBBYTES de la ronde 10. Le motif de propagation étant fixe, il est facile d'identifier l'octet réellement fauté. En effet, on a 6 octets fautés par la même erreur et un octet fauté avec une erreur différente (figure 3.10b) sur le chiffré fauté. L'octet réellement fauté est facilement

identifiable et les six octets fautés donnent une information supplémentaire sur le bit de l'octet fauté (faute mono-bit). Cette information permet de réduire les hypothèses de départ dans l'attaque de Giraud.

De plus la perte d'information liée au SHIFTROWS croisé n'est pas totale. Le croisement entre les deux chemins ne se fait que sur la moitié des bits d'un octet, une moitié d'information reste encore exploitable. Si l'attaque est réalisée en boite blanche (c.à.d avec la connaissance des bits croisés), il est facile d'écarter les bits n'apportant aucune information. Seuls les quatre bits considérés comme apportant une information réelle sont utilisés pour l'attaque. Pour un couple de chiffrés correct/fauté, les hypothèses de sous-clef identifiées comme potentiellement correctes sont plus nombreuses. Pour identifier les bonnes hypothèses de sous-clef, le nombre de couples de chiffrés correct/fauté va donc être plus important que dans le cas d'une attaque classique sans contre-mesure.

La seconde analyse porte sur l'injection d'une faute sur le chemin de données complémentées avant le début de la dernière ronde de l'AES. La contre-mesure va donc détecter la faute puis la propager à travers les deux chemins de données, comme on peut le voir sur la figure 3.17. On retrouve en fin de chiffrement le même motif de propagation que pour une faute injectée sur le chemin de données principales. Comme pour une faute sur le chemin de données principales, on peut identifier l'octet fauté ainsi que le ou les bits fautés.

L'attaque de Giraud est aussi possible avec cette injection de fautes. En effet, le SHIFTROWS croisé permet cette fois-ci d'obtenir de l'information sur l'octet réellement fauté du chemin de données complémentées. On se retrouve alors dans le même cas que lors d'une faute sur le chemin de données normales, seule la moitié des bits de l'octet fauté apporte une réelle information. Pour identifier les bonnes hypothèses de sous-clef, le nombre de couples de chiffrés correct/fauté va donc être plus important que dans le cas d'une attaque classique sans contre-mesure. En revanche il n'est pas possible de savoir si la faute a été injectée dans le chemin de données complémentées ou dans le chemin de données normales.

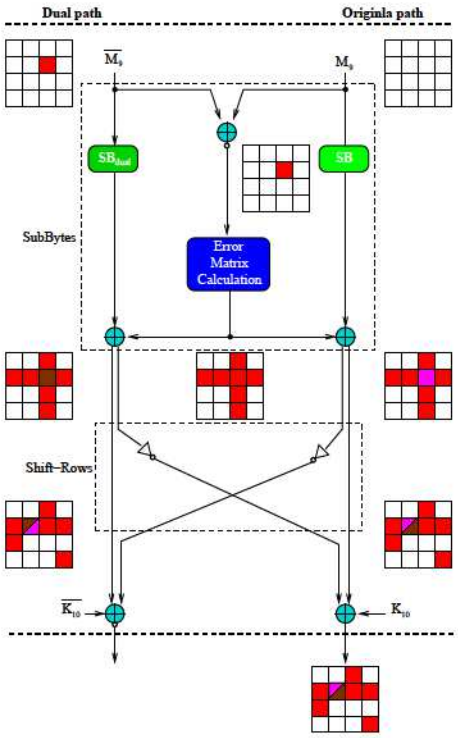


FIGURE 3.17: Propagation d'une faute injectée à travers le chemin de données complémentées lors de la dernière ronde de l'AES.

Injection de fautes sur les deux chemins de données

Si une même faute est injectée sur les deux chemins de données simultanément, lors de la comparaison des deux chemins de données par le mécanisme de détection-propagation, aucune différence ne va être détectée. Les contre-mesures seront mises en défaut. La figure 3.18 illustre cette injection de fautes sur les deux chemins de données simultanément. Les contre-mesures ne pouvant pas détecter les fautes injectées et le SHIFTROWS croisé n'ayant aucun effet sur ces deux fautes, on se retrouve alors dans le même cas qu'un circuit non protégé. Si les fautes sont injectées avant le début de la transformation MIXCOLUMNS de la ronde 9, ou au début de la transformation SUBBYTES de la ronde 10, on peut utiliser respectivement l'attaque de Piret et al. [44] ou l'attaque de Giraud [18] pour retrouver respectivement quatre octets ou un octet de la sous clef.

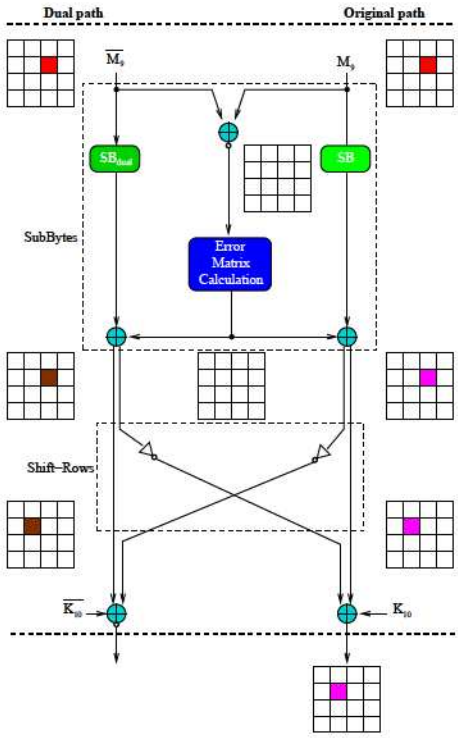


FIGURE 3.18: Propagation de deux fautes identiques injectées à travers les deux chemins de données lors de la dernière ronde de l'AES.

Injection de fautes dans le mécanisme de détection

Un autre point d'entrée pour l'injection de fautes pourrait être le mécanisme de détection lui-même. Ce type d'injection permettrait de neutraliser le croisement des données dans l'opération SHIFTROWS. En effet, si une faute est injectée directement dans le mécanisme de détection, celle-ci va alors être propagée et diffusée de la même manière sur les deux chemins de données. Lors de la transformation SHIFTROWS croisé, les fautes étant identiques sur les deux chemins, aucune perte d'information n'intervient et le croisement est transparent.

Pour pouvoir utiliser ce point d'entrée pour mener une attaque et ainsi retrouver la clef secrète, l'injection de fautes doit intervenir en début de neuvième ronde. Le mécanisme de détection intervient en parallèle de l'opération SUBBYTES, donc si l'injection intervient lors de la dernière ronde, la faute n'affectera pas l'opération SUBBYTES et donc l'attaque de Giraud [18] ne sera pas applicable. En revanche, si une faute est injectée

lors de la ronde 9, la faute va alors affecter la transformation MIXCOLUMNS. L'attaque de Piret et Quisquater [44], présentée dans la partie 3.3.5 peut permettre de retrouver la clef secrète à partir des chiffrés fautés, avec une faute injectée en ronde 9. De même que pour une faute injectée sur un des deux chemins de données, la position de l'octet fauté à travers le mécanisme de détection peut être retrouvée facilement en analysant les positions des octets fautés du chiffré fauté ainsi qu'en suivant le schéma de propagation décrit figure 3.19.

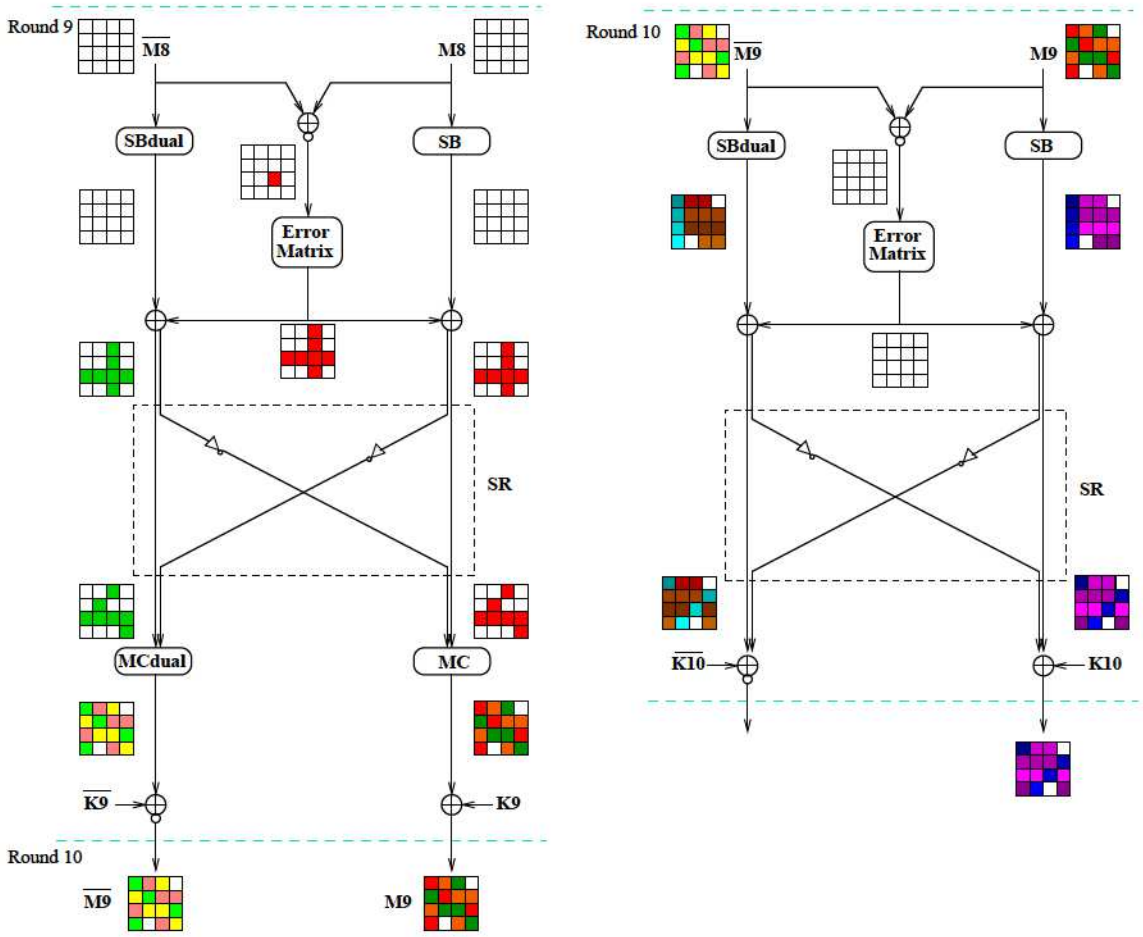


FIGURE 3.19: Propagation d'une faute injectée dans le mécanisme de détection d'erreurs lors de la ronde 9 sur les deux chemins de données.

Lorsqu'une erreur est injectée dans le mécanisme de détection (représentée en rouge), celle-ci est donc propagée sur la ligne et la colonne de la matrice d'état, puis diffusée

sur les deux chemins de données. On a donc avant la transformation MIXCOLUMNS sept octets fautés avec la même valeur. Le MIXCOLUMNS va propager les erreurs sur les colonnes de la matrice d'état, si bien qu'à la fin de la ronde 9, on a 14 octets fautés. Les octets fautés ayant la même couleur correspondent aux octets ayant la même valeur de faute. L'exécution de la ronde 10 s'effectue normalement, sans détection d'erreurs. On obtient à la fin du chiffrement de l'AES, un chiffré avec 14 octets fautés. Les propriétés de la transformation MIXCOLUMNS (présentées partie 3.2) et la position des différents octets fautés dans la matrice d'état avant cette transformation permettent d'obtenir deux octets non fautés en fin de chiffrement. Cette caractéristique peut être utilisée pour détecter les chiffrés fautés correspondant à une faute injectée dans le mécanisme de détection.

En observant la propagation de fautes lors de l'exécution des deux dernières rondes, on remarque qu'une seule colonne de la matrice d'état est affectée par une faute mono-octet en entrée du MIXCOLUMNS de la ronde 9 (la $1^{ère}$ colonne en partant de la gauche sur la figure 3.19), les autres colonnes ayant été affectées par des fautes multi-octets (le SHIFTROWS a décalé la colonne fautée en sortie de la matrice d'erreurs). L'attaque de Piret et Quisquater [44], telle que décrite ne s'appliquant qu'avec des fautes mono-octets, on ne pourra retrouver que quatre octets à la fois. En revanche, l'attaque présentée par Moradi et al. [39] en 2006 est une attaque généralisée utilisant des fautes multi-octets avant le MIXCOLUMNS de la ronde 9 et permet de retrouver la clef secrète. Cette attaque définit un modèle de fautes très large : n'importe quelles fautes avant la transformation MIXCOLUMNS de la ronde 9. Ce modèle autorise donc d'avoir tous les octets de la matrice d'état fautés avant le MIXCOLUMNS de la ronde 9. L'écriture des équations de la propagation des fautes par le MIXCOLUMNS et le SHIFTROWS permet alors de créer des ensembles d'hypothèses de valeurs de sous-clefs. Ces ensembles sont ensuite réduits à l'aide des couples de chiffrés correct/fauté et permettent de retrouver la valeur des différents octets de la sous-clef. Cependant, le nombre de couples de chiffrés correct/fauté nécessaire pour retrouver la valeur correcte des différents octets de la sous-clef peut aller de six couples de chiffrés à plus d'un millier. Dans notre cas, on se retrouve avec 7 octets fautés réparties sur les quatre colonnes de la matrice d'état avant la transformation MIXCOLUMNS de la ronde 9 (*cf.* figure 3.19). Ce type d'attaque permettrait alors de

retrouver la clef secrète avec un nombre réduit de couples de chiffrés correct/fauté par rapport à une attaque de Piret et al.

Injection de fautes dans le mécanisme de diffusion

Une faute peut aussi être injectée dans le mécanisme de diffusion de l'erreur. L'effet recherché est le même qu'avec une faute injectée dans le mécanisme de détection : neutraliser le SHIFTROWS croisé. Mais ici, la faute n'est pas propagée sur la ligne et la colonne relative à sa position dans la matrice d'état, puisque la faute est injectée après le mécanisme de propagation. La figure 3.20 illustre la propagation d'une faute injectée dans le mécanisme de diffusion lors de l'exécution de la ronde 9.

On observe que la faute injectée n'affecte à travers la transformation MIXCOLUMNS de la ronde 9 que la colonne correspondant à sa position d'injection dans la matrice d'état. On se retrouve donc dans le cas simple d'une attaque en fautes sur l'opération MIXCOLUMNS sur un AES non protégée. Cependant, on ne peut retrouver que 4 octets de la sous-clef à la fois contrairement à l'injection dans le mécanisme de détection qui permet de retrouver directement l'intégralité de la clef secrète.

3.6.2 Localisation des blocs SUBBYTES

La logique du circuit étant éparpillée ("scramblée") sur toute la surface du circuit, avant de tenter d'injecter des fautes à différents moments de l'exécution de l'algorithme, il est important de repérer les positions permettant d'injecter des fautes sur chaque octet. Cette première étape est nécessaire afin d'obtenir un gain de temps lors de futures injections de fautes visant à appliquer des attaques en fautes sur ce circuit. En effet, la logique du circuit étant éparpillée, il serait trop long et pas forcément utile de scanner le circuit entier. Certaines zones scannées ne correspondraient pas à l'injection de fautes et le nombre de tentatives d'injections de fautes nécessaires afin d'obtenir suffisamment de couples de chiffrés correct/fauté pour mener une attaque en fautes serait trop important. Cette première phase de détection permet donc d'identifier les zones correspondant à chacun des 16 octets avec seulement quelques chiffrements.

La surface du circuit a été divisée en 36 zones de $125 \times 125 \mu m^2$, correspondant au spot laser le plus large à notre disposition. Pour chaque position, 50 chiffrements

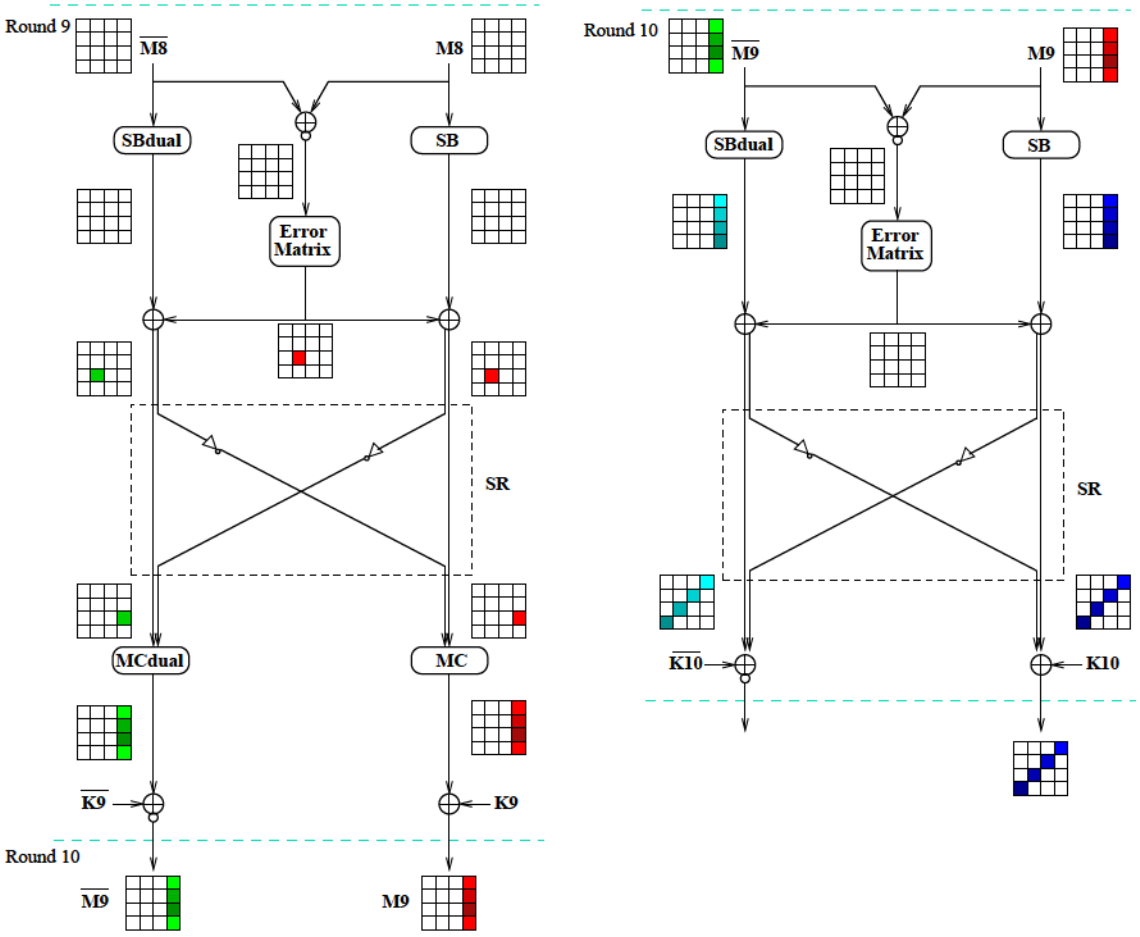


FIGURE 3.20: Propagation d'une faute injectée dans le mécanisme de diffusion d'erreur lors de la ronde 9 sur les deux chemins de données.

sont réalisés avec des textes aléatoires. Pour chaque tir, on vérifie que le chiffré fauté correspond au schéma décrit dans la partie 3.6.1. Cela permet d'identifier quelle zone correspond à quel registre de l'opération SUBBYTES pour chaque octet. Ces zones seront donc utilisées pour injecter des fautes sur les octets choisis dans le but de réaliser les attaques décrites dans la partie précédente 3.6.1.

Les paramètres du laser utilisés pour les différentes injections sont une puissance de 750 nJ par tir avec une longueur d'onde de 532 nm. Avec le coefficient de transmission de l'optique, on obtient une densité d'énergie de 17 pJ/μm^2. La durée de pulse du laser est fixe et égale à 5 ns.

La Figure 3.21 montre la surface de l'ASIC AES avec les 36 zones ainsi que l'identification de 14 zones correspondant aux registres des 16 octets. Pour chaque zone est identifié le ou les numéros des octets correspondant.

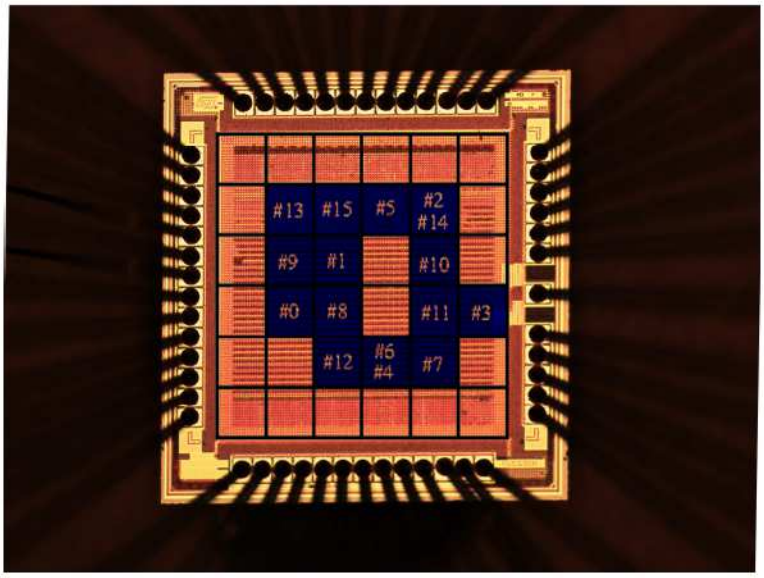


FIGURE 3.21: Identification de plusieurs zones correspondant aux registres du SUBBYTES pour différents octets.

Après identification des registres correspondant à tous les octets, il apparaît que la disposition du circuit est en couronne. En effet les parties extérieures et centrales de l'ASIC ne correspondent à aucun registre du bloc SUBBYTES. Plus de la moitié de la surface du circuit ne permet pas d'injecter de fautes. Il auraient donc été inutile de scanner ces zones lors de campagnes d'injections de fautes pour mener différentes attaques.

3.6.3 Résultats

Attaque sur la transformation SUBBYTES

Avec les chiffrés fautés correspondant à une injection de fautes dans le registre de la transformation SUBBYTES du chemin de données normales, lors de l'exécution de la dernière ronde, une première attaque fut menée en boîte noire, sans connaissance de l'architecture du mécanisme de détection-propagation de l'erreur ni de celle du SHIFTROWS

croisé. S'il est facile de trouver le schéma de propagation (fixe) de la contre-mesure, l'absence de connaissance sur l'architecture du SHIFTROWS croisé est un point bloquant. Le tableau 3.8 montre le résultat d'un chiffré correct, le chiffré fauté avec une faute mono-bit en début de dernière ronde ainsi que les fautes associées à ce couple de chiffrés.

TABLE 3.8: Exemple de chiffrés correct/fauté.

Chiffré correct	1C 4A 0F B9 1D 9E AD FB 4F 23 3A 7B D9 50 CC 29
Chiffré fauté	1C 4A 0F F5 1D 9E A9 FF 4F 27 3A 7F DD 50 CC 2D
Erreur	00 00 00 4C 00 00 04 04 00 04 00 04 04 00 00 04

En appliquant la transformation SHIFTROWS inverse, on retrouve bien le schéma de propagation de faute lorsque celle-ci est injectée en début de dernière ronde (6 valeurs de fautes identiques et 1 valeur de faute unique). On peut vérifier que l'on a bien injecté une faute mono-bit (la valeur de faute propagée est 0x04) sur l'octet 16. L'attaque de Giraud peut donc s'appliquer sur cet octet fauté. Néanmoins, les informations perdues lors du SHIFTROWS croisé (croisement des bits 3, 4, 5 et 7 pour l'octet 16) n'ont pas permis de converger vers la bonne hypothèse de sous-clef pour cette octet même si un apprentissage pourrait être effectué par l'attaquant pour retrouver les croisements du SHIFTROWS croisé .

Lorsque l'attaque fut menée en boîte blanche, avec la connaissance de l'implémentation du SHIFTROWS croisé, il était facile de mettre en place un filtre permettant de trier les bits de l'octet réellement fauté et de seulement garder les bits comportant l'information réelle sur la faute injectée. En reprenant l'exemple précédent de l'octet 16, on a seulement considéré les bits 0, 1, 2 et 6 comme apportant une réelle information et permettant de réduire l'ensemble des hypothèses de la sous-clef. Pour cet octet, 6 couples de chiffrés correct/fauté ont été nécessaires pour obtenir la valeur de l'octet de la sous-clef. En appliquant cette attaque aux différents couples de chiffrés correct/fauté à notre disposition, plusieurs octets de la sous-clef furent donc retrouvés.

Attaque du mécanisme de détection-propagation de fautes

Les tentatives d'injections de fautes dans le mécanisme de détection-propagation de fautes, que ce soit pour injecter une faute dans le mécanisme de détection ou dans le

mécanisme de propagation, n'ont pas donné de résultats exploitables. Aucune faute n'a pu être injectée dans ces deux mécanismes.

Plusieurs raisons peuvent expliquer cet échec. La première raison est que ce mécanisme, pour son implémentation matérielle, n'utilise aucun registre. La contre-mesure est intégralement implémentée en portes logiques. Cela nécessite une synchronisation parfaite entre l'exécution de l'algorithme de chiffrement par l'ASIC et le tir laser. Notre banc laser ayant une gigue de 10 ns, la précision de synchronisation nécessaire pour ce type d'injection de fautes dans la logique n'a probablement pas pu être obtenue.

La seconde explication possible à l'absence d'injection de fautes via cette contre-mesure est le recouvrement possible des portes logiques, ou du moins des zones sensibles de ces portes, constituant la contre-mesure, par les différentes couches de métallisation. Les tirs laser étant effectués par la face avant, le faisceau laser a pu être réfléchi par les couches de métallisation, ne permettant pas d'atteindre les zones sensibles de ces portes.

De même aucune faute n'a pu être injectée simultanément sur les deux chemins de données pour pouvoir rendre inefficace les contre-mesures du circuit. Une cause possible de cet échec est la dispersion de la logique du circuit. En effet, le laser ayant un effet local, la même faute ne peut être injectée sur un octet de chaque chemin de données si les registres correspondant à ces octet sont trop éloignés l'un de l'autre. Cependant, ce type d'injection a pu être réalisé à l'aide d'injection de fautes par glitchs d'horloge [2].

3.6.4 Conclusion et préconisations

Cette étude a permis de mettre à l'épreuve les contre-mesures implémentées dans un ASIC AES contre les injections de fautes et de pouvoir évaluer l'efficacité de celles-ci.

Pour les attaques en boîte noire, les attaques classiques n'ont pas réussi. Cela est dû principalement à la contre-mesure du SHIFTROWS croisé qui implique la perte d'une partie ou de l'intégralité de l'information, dépendant du type de fautes injectées (mono-bit ou multi-bits). Néanmoins, lorsque l'attaque est menée en boîte blanche, les octets de la clef de ronde sont retrouvés. Ce résultat était attendu. La faiblesse de la contre-mesure en boîte blanche avait déjà été identifiée [23]. Cette partie de l'étude a permis de confirmer cette faiblesse liée à l'architecture du mécanisme de détection-propagation et à l'absence de caractère aléatoire dans les fautes diffusées.

L'autre objectif de cette étude était d'identifier d'autres vulnérabilités éventuelles du circuit aux attaques en fautes. Dans cette optique, il a été identifié que le mécanisme lui-même peut être utilisé pour injecter des fautes et ainsi rendre inefficace d'autres contre-mesures (ici le SHIFTROWS croisé). Cependant, cette vulnérabilité n'est restée qu'à l'état théorique puisqu'aucune faute n'a pu être injectée dans le mécanisme de détection.

Ces résultats permettent de confirmer la nécessité d'injecter de l'aléatoire lorsque l'on désire propager une faute après détection dans les chemins de données dans un but de confusion de l'attaquant. L'aléatoire peut être introduit de plusieurs manières, par exemple dans le schéma de propagation de la faute détectée, empêchant ainsi l'attaquant d'obtenir une information sur la localisation de la faute injectée et rendant l'analyse pour retrouver la clef beaucoup plus longue. Une autre voie pour augmenter l'efficacité de la contre mesure est de propager non pas la faute injectée, mais une valeur aléatoire, rendant les attaques de type *DFA* inefficaces. Cependant, même en utilisant des valeurs aléatoires, l'ASIC AES reste vulnérable aux attaques de type *Safe Error* (*c.f.* partie 3.3.6). L'utilisation de code correcteur d'erreur peut permettre de se prémunir contre ce type d'attaque en fautes. En plus d'être détectée, l'erreur est ensuite corrigée, l'attaquant n'a alors aucune information sur le fait d'avoir injecté une faute ou pas.

Conclusion générale

L'objectif de cette thèse était d'étudier les modèles de fautes induits par les tirs laser sur les circuits cryptographiques et d'exploiter ces modèles afin d'évaluer les menaces liées aux attaques par fautes vis-à-vis de ces circuits cryptographiques.

Tout d'abord un état de l'art des différents effets d'un tir laser sur un circuit CMOS a été effectué, ce qui a permis de mettre en évidence les mécanismes d'injection d'une faute dans un circuit intégré soumis à un tir laser ainsi que les modèles de fautes associés. Par la suite, une étude de l'influence des différents paramètres d'un tir laser a aussi été réalisée afin d'avoir une vue complète de l'injection de fautes par un tir laser que ce soit lors d'injections de fautes statiques ou dynamiques .

La première étude des modèles de fautes injectées par un tir laser fut réalisée sur un circuit de test intégrant un point mémoire de type SRAM. Les résultats expérimentaux ont montré l'absence de faute de type *Bit-flip* contrairement à ce qu'une analyse théorique du layout du point mémoire prenant en compte la taille du spot laser semblait indiquer. À la suite de ces premiers résultats expérimentaux, des simulations électriques SPICE ont été menées pour comprendre l'absence de fautes de type *Bit-flip*. Ces simulations ont mis en évidence l'influence du layout lors d'un tir laser et l'apparition ou

non de fautes. La corrélation entre les résultats expérimentaux et les simulations SPICE a montré que le modèle de faute *Bit-flip* n'était pas pertinent, contrairement à ce que semblaient indiquer les hypothèses théoriques.

Pour confirmer ces résultats expérimentaux et de simulations, des injections de fautes ont été effectuées sur la mémoire RAM d'un micro-contrôleur 8-bits. Une zone de cette mémoire RAM contenant plusieurs cellules SRAM fut cartographiée en conservant les paramètres de tir utilisés pour les injections de fautes effectuées sur le point mémoire SRAM. De même que pour la cellule SRAM du circuit test, le modèle de fautes *Bit-flip* n'est pas pertinent. En revanche, lorsque la durée du tir laser est fixée à 30 ps, certaines fautes de type *Bit-flip* apparaissent (de l'ordre de 1% du total des fautes injectées). Les fautes de type *Bit-reset* ou *Bit-set* restent néanmoins prépondérantes.

Pour les tirs lasers sur cellule mémoire de type SRAM, il est donc plus pertinent de considérer le modèle de fautes *Bit-set* ou *Bit-reset* que le modèle *Bit-flip* même si celui-ci reste faisable.

Dans un deuxième temps, un ASIC implémentant l'algorithme de chiffrement AES a été utilisé pour étudier les modèles de fautes lors de tir laser par la face avant avec un spot de grande taille. Les résultats ont montré que les deux modèles de fautes étaient pertinents. La présences des couches métalliques du circuit intégré lors du tir laser peuvent avoir eu une influence notable sur l'injection de fautes associée aux deux modèles considérés. L'utilisation d'un spot laser de grande taille a aussi permis d'obtenir des fautes mono-bit. Les différentes couches de métallisation considérées comme une contrainte pour l'injection de fautes ont pu être un atout pour obtenir des fautes précises avec un dispositif d'injection à faible coût.

De plus, les données collectées ont permis de conduire un comparatif de plusieurs attaques de la littérature au vu des types de fautes obtenues mais aussi de simplifier en terme de calcul nécessaire une attaque pour ces types de fautes.

Enfin, la connaissances des modèles de fautes possibles sur ce circuit ASIC a permis de conduire une évaluation des contres-mesures embarquées sur ce circuit intégré et de confirmer plusieurs hypothèses concernant les faiblesses de ces contre-mesures.

Pour approfondir ces différents résultats, des injections laser ainsi que des simulations pourront être menées sur des cellules mémoires SRAM afin de mieux comprendre l'effet de la durée de pulse d'un tir laser sur l'injection de fautes ainsi que les modèles de fautes associés. Cette thèse n'a pas traité des fautes injectées dans des portes logiques. Il serait donc intéressant d'étudier la propagation d'une faute dans la logique ainsi que les modèles de fautes induits par un tir laser, puis de comparer avec les modèles de fautes induits sur une cellule SRAM.

Bibliographie

[1] M. Agoyan, J.-M. Dutertre, A.-P. Mirbaha, D. Naccache, A.-L. Ribotta, and A. Tria. How to flip a bit ? In *On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International*, pages 235–239, 2010. Cité pages 11 and 37.

[2] M. Agoyan, J.M. Dutertre, D. Naccache, B. Robisson, and A. Tria. When clocks fail : On critical paths and clock faults. *Smart Card Research and Advanced Application*, pages 182–193, 2010. Cité pages 10, 11, and 119.

[3] M. Agoyan, S. Bousquet, J.-M. Dutertre, J. Fournier, J.-B. Rigaug, B. Robisson, and A. Tria. Design and Characterisation of an AES chip embedding countermeasures. In *International Journal of Intelligent Engineering Informatics 1*, volume 3-4, pages 328–347, 2011. Cité pages 88 and 89.

[4] M.A. Bajura, Y. Boulghassoul, R. Naseer, Sandeepan Das Gupta, A.F. Witulski, J. Sondeen, S.D. Stansberry, J. Draper, L.W. Massengill, and J.N. Damoulakis. Models and Algorithmic Limits for an ECC-Based Approach to Hardening Sub-100-nm SRAMs. *Nuclear Science, IEEE Transactions on*, 54(4) :935–945, 2007. Cité page 49.

[5] A. Balasubramanian, D. McMorrow, S.A. Nation, B.L. Bhuva, R.A. Reed, L.W.

Massengill, T.D. Loveless, O.A. Amusan, J.D. Black, J.S. Melinger, M.P. Baze, V. Ferlet-Cavrois, M. Gaillardin, and J.R. Schwank. Pulsed Laser Single-Event Effects in Highly Scaled CMOS Technologies in the Presence of Dense Metal Coverage. *Nuclear Science, IEEE Transactions on*, 55(6) :3401 –3406, dec. 2008. ISSN 0018-9499. doi : 10.1109/TNS.2008.2007295. Cité pages 27 and 108.

[6] A. Barenghi, G. Bertoni, L. Breveglieri, M. Pellicioli, and G. Pelosi. Low Voltage Fault Attacks to AES. In *HOST*, pages 7–12, 2010. Cité pages 10 and 11.

[7] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525, 1997. Cité page 79.

[8] J. Blömer and J.-P. Seifert. Fault Based Cryptanalysis of the Advanced Encryption Standard (AES). In *Computer Aided Verification*, volume 2742 of *Lecture Notes in Computer Science*, pages 162–181, 2003. Cité pages 22, 72, and 87.

[9] D. Boneh, R.A. DeMillo, and R.J. Lipton. On the importance of checking cryptographic protocols for faults. In *EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51, 1997. Cité page 79.

[10] F. Darracq, H. Lapuyade, N. Buard, F. Mounsi, F. Foucher, P. Fouillat, M-C. Calvet, and R. Dufayel. Backside SEU Laser Testing for Commercial Off-The-Shelf SRAMs. *IEEE Transactions on Nuclear Science*, Vol 49(6) :pp 2977–2983, December 2002. Cité pages vii, 28, and 31.

[11] A. Dehbaoui, J.-M. Dutertre, B. Robisson, and A. Tria. Electromagnetic Transient Faults Injection on a Hardware and Software Implementation of AES. In *9th Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2012. Cité pages 10 and 11.

[12] W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, pages 644–654, 1976. Cité page 8.

[13] A. Douin, V. Pouget, F. Darracq, D. Lewis, P. Fouillat, and P. Perdu. Influence of Laser Pulse Duration in Single Event Upset Testing. *IEEE Transactions on Nuclear Science*, Vol 53(4) :pp 1799–1805, August 2006. Cité pages vii, 28, 29, 30, and 64.

[14] J.-M. Dutertre, A.-P. Mirbaha, D. Naccache, A.-L. Ribotta, A. Tria, and T. Vaschalde. Fault Round Modification Analysis of the advanced encryption standard. In *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*, pages 140–145, 2012. Cité page 79.

[15] J.M. Dutertre, F.M. Roche, P. Fouillat, and D. Lewis. Improving an SEU Hard Design using a Pulsed Laser. 2001. Cité page 22.

[16] B.M. GAMMEL and S. MANGARD. On the Duality of Probing and Fault Attacks. In *J.Electron.Test.*, volume 26, pages 483–493, August 2010. Cité pages 10 and 11.

[17] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis : concrete result. In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261, 2001. Cité pages 10 and 11.

[18] C. Giraud. DFA on AES. In *Advanced Encryption Standard – AES*, volume 3373 of *Lecture Notes in Computer Science*, pages 571–571, 2005. Cité pages ix, 79, 80, 81, 104, 108, 109, 111, and 112.

[19] D. H. Habing. The Use of Lasers to Simulate Radiation-Induced Transients in Semiconductor Devices and Circuits. In *Nuclear Science, IEEE Transactions on*, volume 12, pages 91 –100, 1965. doi : 10.1109/TNS.1965.4323904. Cité page 12.

[20] H. HANDSCHUH, P. PAILLIER, and J. STERN. Probing Attacks on Tamper-Resistant Devices. In *First International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 1999. Cité pages 10 and 11.

[21] A.H. Johnston. Charge Generation and Collection in p-n Junctions Excited with Pulsecd Infrared Laser. *IEEE Transactions on Nuclear Science*, Vol 40(6) :pp 1694–1702, December 1993. Cité page 24.

[22] M. Joye and M. Tunstall, editors. *Fault Analysis in Cryptography.* Springer, 2012. Cité page 79.

[23] M. Joye, P. Manet, and J-B Rigaud. Strengthening hardware AES implementations against fault attacks. *Information Security, IET*, 1(3) :106–110, 2007. Cité pages 89 and 119.

[24] David Kahn. Codebreakers : L'histoire de l'écriture secrète. 2008. Cité page 7.

[25] Auguste Kerckhoffs. La cryptographie militaire. In *Journal des sciences militaires*, pages 161–191, 1883. Cité page 7.

[26] Chong Hee Kim. Improved Differential Fault Analysis on AES Key Schedule. *Information Forensics and Security, IEEE Transactions on*, 7(1) :41–50, 2012. Cité page 79.

[27] Neal Koblitz. Elliptic Curve Cryptosystems. 48(177) :203–209, 1987. Cité page 8.

[28] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology — CRYPTO' 99*, volume 1666 of *Lecture Notes in Computer Science*, pages 789–789, 1999. Cité pages 10 and 11.

[29] F. Koeune and F.-X. Standaert. A Tutorial on Physical Security and Side-Channel Attacks. In *Foundations of Security Analysis and Design III*, volume 3655 of *Lecture Notes in Computer Science*, pages 78–108, 2005. Cité pages 10 and 11.

[30] O. Kömmerling and M. G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. In *Proceedings of the USENIX Workshop on Smartcard Technology*, pages 9–20, 1999. Cité pages 10 and 11.

[31] R. Lashermes, G. Reymond, J.-M. Dutertre, J. Fournier, B. Robisson, and A. Tria. A DFA on AES based on the Entropy of Error Distributions. In *9th Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2012. Cité pages ix, 79, 83, 85, and 108.

[32] D. Lewis, V. Pouget, F. Beaudoin, P. Perdu, H. Lapuyade, P. Fouillat, and A. Touboul. Backside Laser Testing of ICs for SET Sensitivity Evaluation. *IEEE*

Transactions on Nuclear Science, Vol 48(6) :pp 2193–2201, December 2001. Cité pages vii, 25, 27, and 28.

[33] P. Loubet-Moundi, D. Vigilant, and F. Olivier. Static Fault Attacks on Hardware DES Registers. Cryptology ePrint Archive, Report 2011/531, 2011. Cité page 22.

[34] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2) :373–386, 1988. Cité page 9.

[35] V. Maingot and R. Leveugle. Analysis of Laser-Based Attack Effects on a Synchronous Circuit. In *2nd International Design and Test Workshop(IDT)*, 2007. Cité page 22.

[36] J.S. Melinger, S. Buchner, D. McMorrow, W.J. Stapopr, T.R. Weatherford, A.B. Campbell, and H.Eisen. Critical Evaluation of the Pulsed Laser Method for Single Event Effects Testing and Fundamental Studies. *IEEE Transactions on Nuclear Science*, Vol 41(6) :pp 2574–2583, December 1994. Cité pages vii, 24, and 25.

[37] MICROPACKS. http ://www.arcsis.org/micropacks.html. Cité page 93.

[38] Victor Miller. Use of Elliptic Curves in Cryptography. In *CRYPTO*, pages 417–426, 1985. Cité page 8.

[39] A. Moradi, M. T. Manzuri Shalmani, and M. Salmasizadeh. A Generalized Method of Differential Fault Attack Against AES Cryptosystem. In *CHES*, pages 91–100, 2006. Cité pages 79 and 114.

[40] NIST. http ://www.csrc.nist.gov/. Cité page 9.

[41] NIST. Data Encryption Standard (DES). Federal Information Processing Standards Publication, n. 46, 3, Reaffirmed October 25, 1999. Cité page 9.

[42] NIST. Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication, n. 197, November 26, 2001. Cité pages 9 and 75.

[43] D. Otto. *Fault Attacks and Countermeasures*. PhD thesis, Paderborn University (Germany), 2004. Cité page 22.

[44] G. Piret and J.-J. Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad. In *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 77–88, 2003. Cité pages ix, 79, 86, 111, 113, and 114.

[45] V. Pouget, P. Fouillat, D. Lewis, H. Lapuyade, L. Sarger, F.M. Roche, S. Duzellier, and R. Ecoffet. An overview of the applications of a pulsed laser system for SEU testing. In *6th IEEE International On-Line Testing Workshop (IOLTS)*, pages 52 –57, 2000. doi : 10.1109/OLT.2000.856612. Cité page 22.

[46] V. Pouget, A. Douin, G. Foucard, P. Peronnard, D. Lewis, P. Fouillat, and R. Velazco. Dynamic Testing of an SRAM-based FPGA by Time-Resolved Laser Fault Injection. In *14th IEEE International On-Line Testing Workshop (IOLTS)*, 2008. Cité page 22.

[47] Matthieu Rivain. Differential Fault Analysis on DES Middle Rounds. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 457–469, 2009. Cité page 83.

[48] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In *Communications of the ACM*, volume 21, pages 120–126, 1978. Cité page 8.

[49] T. Roche, V. Lomné, and K. Khalfallah. Combined Fault and Side-Channel Attack on Protected Implementations of AES. In *Smart Card Research and Advanced Applications*, pages 65–83, 2011. Cité pages ix, 79, 81, 82, 83, and 108.

[50] C. Roscian, J.-M. Duterte, and A. Tria. Injection de fautes laser et localisation de blocs logiques. In *CRYPTO'PUCES 2011*. Institut de Mathématiques de Luminy, Mai 2011. Cité page 73.

[51] C. Roscian, J.-M. Duterte, and A. Tria. Frontside Laser Fault Injection on Cryptosystems - Application to the AES last round. In *IEEE Int. Symposium on Hardware-Oriented Security and Trust (HOST)*, 2013. Cité page 73.

[52] C. Roscian, A. Sarafianos, J.-M. Dutertre, and A. Tria. Discussion on the Model of Laser-Induced Faults in SRAM Memory Cells. In *Forth International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*, March 2013. Short talk session. Cité page 35.

[53] C. Roscian, A. Sarafianos, J.-M. Dutertre, and A. Tria. Fault Model Analysis of Laser-Induced Faults in SRAM Memory Cells. In *Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2013. Cité page 35.

[54] C.l Roscian, F. Praden, J.-M. Dutertre, J. Fournier, and A. Tria. Security Characterisation of a Hardened AES Cryptosystem Using a Laser. *TECHNICAL SCIENCES, University of Warmia and Mazury publishing*, No 15(1), July 2012. Cité page 73.

[55] Cyril Roscian. New Fault Analysis of the AES Cryptosystem : Experimental Validation with Laser. In *International Workshop on Practical Hardware Innovations in Security Implementation and Characterisation*. Secure Architectures and Systems Department, École des mines de Saint-Étienne, October 2011. Poster Session. Cité page 73.

[56] A. Sarafianos, O. Gagliano, M. Lisart, V. Serradeil, J.M. Dutertre, and A. Tria. Electrical modeling of the photoelectric effect induced by a pulsed laser applied to an NMOS transistor. In *IEEE International Reliability Physics Symposium (IRPS)*, 2013. Cité pages 49, 50, and 51.

[57] A. Sarafianos, C. Roscian, J.-M. Duterte, M. Lisart, O. Gagliano, V. Serradeil, and A. Tria. Robustness improvement of an SRAM cell against laser-induced fault injection. In *16th IEEE Symp. Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, 2013. In Submission. Cité pages 35 and 72.

[58] A. Sarafianos, C. Roscian, J.-M. Dutertre, M. Lisart, and A. Tria. Electrical modeling of the photoelectric effect induced by a pulsed laser applied to an SRAM cell. In *24th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis (ESREF)*, 2013. Cité page 35.

[59] A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, and J.-P. Seifert. Simple Photonic Emission Analysis of AES. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 41–57. Springer Berlin Heidelberg, 2012. Cité page 11.

[60] A. Sedra and K. Smith. Microelectronics Circuits : Fifth Edition. 2004. Cité page 18.

[61] N. Selmane, S. Guilley, and J.-L. Danger. Practical Setup Time Violation Attacks on AES. In *EDCC-7 '08 : Proceedings of the 2008 Seventh European Dependable Computing Conference*, pages 91–96, Washington, DC, USA, 2008. IEEE Computer Society. Cité pages 10 and 11.

[62] S. Skorobogatov and R. Anderson. Optical Fault Induction Attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 31–48, 2003. Cité page 11.

[63] D. Stinson. *Cryptography : Theory And Practice*. Chapman & Hall, third edition edition, 2005. Cité page 10.

[64] M. Tunstall and H. Choukri. Round Reduction Using Faults. In *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 13–24, 2005. Cité page 79.

[65] R. Velazco and F. J. Franco. Single Event Effects on Digital Integrated Circuits : Origins and Mitigation Techniques. 2007. Cité page 22.

[66] F. Wang and V.D. Agrawal. Single Event Upset : An Embedded Tutorial. In *Proc. of 21st International Conference on VLSI Design*, pages pp. 429–434, 2008. Cité page 15.

[67] J. Wolkerstorfer, E. Oswald, and M. Lamberger. Towards An ASIC Implementation of the AES S-BOX. In Springer, editor, *The Cryptographers' Track at the RSA Conference – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 67–78, 2002. Cité pages ix, 76, 89, and 90.

[68] S.M. Yen and M. Joye. Checking before output may not be enough against fault-based cryptanalysis. *IEEE Transactions on Computers*, 49 :967–970, 2000. Cité page 87.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2013 EMSE 0708

Cyril ROSCIAN

Physical cryptanalysis of security chip using LASER sources

Speciality : Microelectronic

Keywords : LASER, Fault injection, DFA, Fault models, AES, SRAM

Cryptographic circuits, because they contain confidential informations, are subject to fraud from malicious users, commonly known as attacks. Several attacks have been published and analysed. One of the most effective attack, called Differential Fault Analysis (DFA), uses some fault, voluntary injected by the attacker during the computations, for example with a laser. However, fault models used by these attacks can be restrictive and determine the effectiveness of the attack. Thus, it is important to know which fault model is useful or feasible according to the targeted device or injection means (in our case the laser).
A first study about the injected fault types (*bit-set*, *bit-reset* or *bit-flip*) on SRAM memory cells highlighted the strong data dependency of the injected faults and the irrelevance of the *bit-flip* fault type. This last result allows to mount *Safe Error* attacks and creates a real security issue. These results were obtain thanks to sensitivity laser map performed on an isolated SRAM cell and on an 8-bits micro-controller RAM memory. To confirm these experimental results, SPICE simulations have been made with a model developed in the department. This model takes into account the topology of the target. Tests were then carried out on an ASIC implementing the AES algorithm. The fault analysis showed the presence of the three types of faults but also a low injection rates. In contrast, the error repeatability was particularly high. This allowed us to simplify an existing attack and to obtain an attack more effective than conventional attacks, requiring fewer faulted cipher text and reducing the complexity of the analysis to find the secret key. Finally, an assessment of the countermeasure of this circuit shown their ineffectiveness with respect to fault laser attacks. Areas for improvement were then proposed.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2013 EMSE 0708

Cyril ROSCIAN

Cryptanalyse physique de circuits sécuritaires à l'aide de sources LASER

Speciality : Microélectronique

Keywords : LASER, Injections de fautes, DFA, Modèles de fautes, AES, SRAM

Les circuits cryptographiques, parce qu'ils contiennent des informations confidentielles, font l'objet de manipulations frauduleuses, appelées communément attaques, de la part de personnes mal intentionnées. Plusieurs attaques ont été répertoriées et analysées. L'une des plus efficaces actuellement, appelée cryptanalyse DFA (Differential Fault Analysis), exploite la présence de fautes, injectées volontairement par l'attaquant par exemple à l'aide d'un laser, dans les calculs. Cependant, les modèles de fautes utilisés dans ces attaques sont parfois très restrictifs et conditionnent leur efficacité. Il est donc important de bien connaître quel modèle de faute est pertinent ou réalisable en fonction du circuit cible et du moyen d'injection (dans notre cas le laser).
Un première étude portant sur le type de fautes (*bit-set*, *bit-reset* ou *bit-flip*) injectées sur des points mémoires SRAM a mis en évidence la forte dépendance des fautes injectées vis à vis des données manipulées et la quasi inexistence de fautes de type *bit-flip*. Ce dernier résultat favorise grandement les attaques de type *Safe Error* et engendre donc un réel problème de sécurité. La mise en évidence de tels résultats a été possible grâce à des cartographies de sensibilité au laser réalisées sur une cellule SRAM isolée puis sur la mémoire RAM d'un micro-contrôleur 8 bits. Pour confirmer ces résultats expérimentaux, des simulations SPICE d'injection de fautes laser ont été réalisées à partir d'un modèle développé dans l'équipe. Ce modèle prend en compte la topologie de la cible. Des tests ont ensuite été réalisés sur un circuit ASIC implémentant l'algorithme AES. L'analyse des fautes a montré la présence des trois types de fautes mais aussi un faible taux d'injection. En revanche, le taux de répétabilité des fautes était particulièrement élevé. Cela nous a permis d'améliorer une attaque existante et d'obtenir au final une attaque plus efficace que les attaques classiques, nécessitant moins de chiffrements fautés et une analyse des résultats réduite pour retrouver la clef secrète. Enfin, une évaluation des contre-mesures embarquées dans ce circuit a montré leurs inefficacité vis à vis des attaques en fautes par laser. Des pistes d'amélioration ont ensuite été proposées.